Artificial Intelligence Neural Network

Software Version: 0.0.2

Paper by:

Dexter Rio Shepherd

Problem

In the 21st century humans, lead busy lives. You don't have time to clean the house as you have to work and when you get home you have to cook. We get little time to ourselves.

If we are too busy and do not get enough rest research shows, we are more likely to have strokes and suffer breakdowns. Living in a messy house is also no way to live and can prevent you resting. Can we really be bothered to clean the house after a long day of work?

Elderly people need care, as do disabled people. When there is no family, around they need a carer but there are not enough carers around to cater to the persons need.

How can we do all these complicated jobs when we are too busy to do it, don't have enough people to do it and don't have the money for a servant?

The system in this paper is a learning algorithm, which learns specifically what the user teaches it. The system can apply its knowledge in similar situations. This system can help millions of people by making their lives easier.

The system can do whatever you can do as it learns through interaction. You can come home to a tidy home and a nice cooked meal every day so you can put your feet up and not have to worry about anything.

Another issue is that current AI on the internet is that people cannot teach it, cleverbot doesn't know who you are, how can you have a conversation with something which takes limited data in.

Planning and Design

Task

The system will need to be able to read a main file and create files and pathway links depending on what is in the input. The system must be able to expand its knowledge through a series of text files with links to each other.

The system will have speech to text for its main input. By using serial communication, the AI can receive and send data to an Arduino; this Arduino will have servos which can read its own position so I can record actions and play them on command. The data from the servos will store on a local SD card.

The AI will need audio output text to speech on top to make it a full audio I/O system. The system will have a graphical interface which allows you to interact.

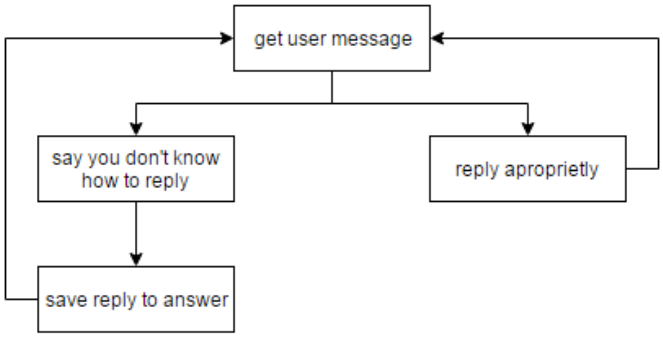
The system will have a data file organized alphabetically with every word in the English language and what they categorize as.

This system will be converted so it can run on raspberry pi and desktop.

Requirement

Take in a user input, find it in its data and output the response. If it is not present in the data it will check what the trigger words are and subjects are, if the trigger word is not found you may also add that to the trigger data. The system will find its way through and then let you tell it how to respond.

User Interface



Success Criteria

I will know the system works when it can do everything my task plan says.

Test plan

Test Number	Test	Expected result
1	The GUI boots up and works	The face image is in the background.
2	The microphone receives input on demand	When you press the microphone button it listens for your voice
3	If the system does not understand, it gives you a message.	A pop up screen comes up.
4	The system outputs your message and the AI's message on screen and stores a log of conversations.	A large string of user message and robot message gets larger and larger on the screen.
5	The system searches its data for an answer	If it is in there it outputs the answer, if not it learns.
6	The system gives out audio output	Uses text to speech to say appropriate words.
7	The system lets you copy the conversation using a function on the menu.	When you go to edit > copy it will copy the conversation so you can paste it later.
8	The menu gives information about the AI and how to use it.	When you go to the help bar it gives options which give pop up bars.
9	If the user presses the copy function they can paste the conversation	Go to the option in edit, and when you select it I will copy the text for you to paste.
10	The user can add trigger words	Go to the option in edit, and when you select it a bar will come up which allows you to enter the data. This data will enter into files.
11	The user can add subject words	Go to the option in edit, and when you select it a bar will come up which allows you to enter the data. This data will

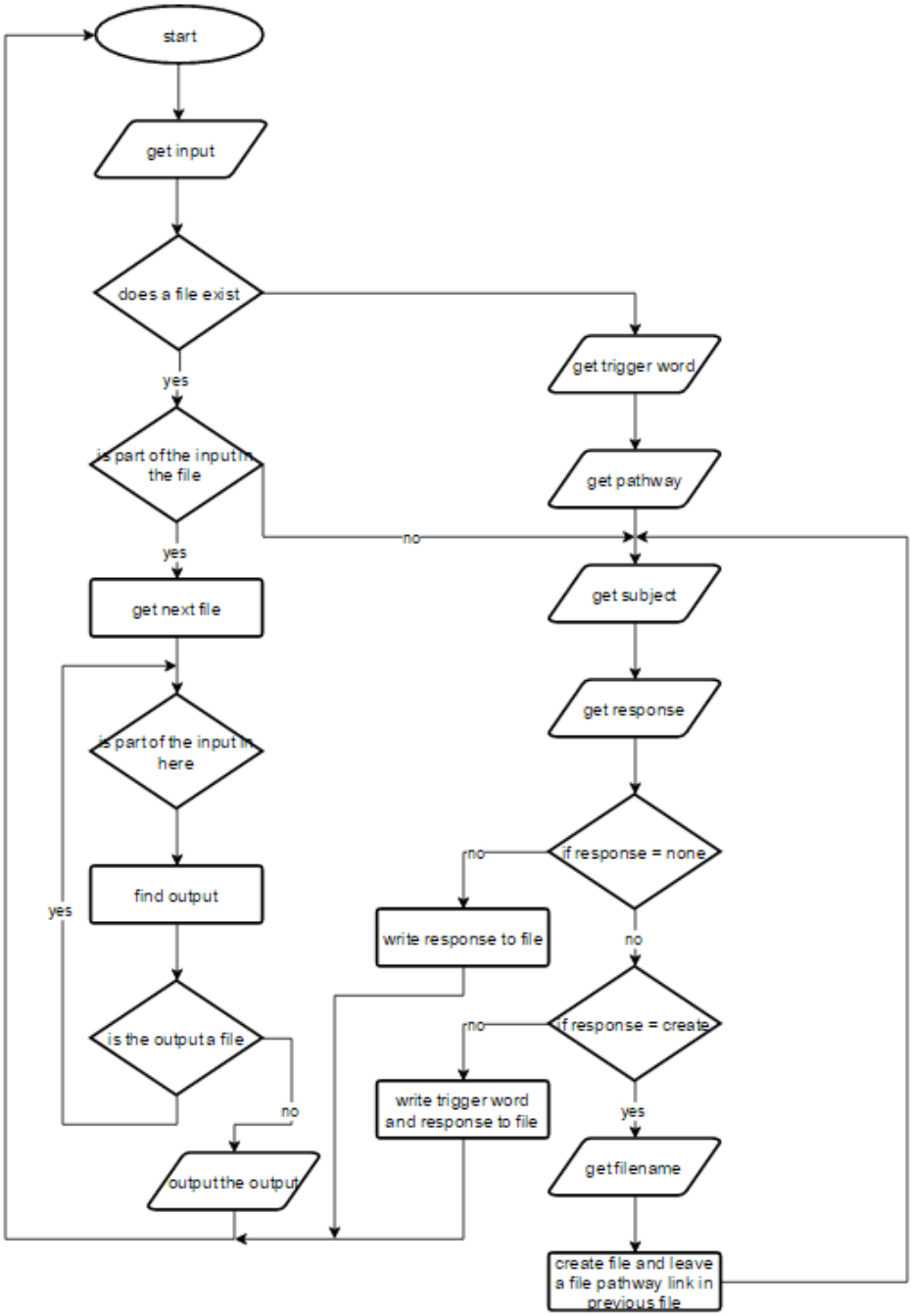
		enter into files.
12	The user can add command words	Go to the option in edit, and when you select it a bar will come up which allows you to enter the data. This data will enter into files.
13	The system can have long strings entered in and pick out keywords	If I add “how ru” and say “how the hell ru” it has the same outcome.
14	The system can communicate with an Arduino to move the system around to make it mobile. Servo learning system will be preferred.	When I say action in the learning system it will send transmissions to the Arduino
15	The AI can learn a series of actions categorized under one action.	The AI will be taught individual things, then taught to do one big action including all of this.
16		

Hardware

1. Raspberry Pi (B)
2. 5” touchscreen display
3. External USB sound card
4. Microphone
5. Speaker
6. Arduino mega
7. Servos with feedback enabled
8. WIFI dongle
9. Power cable
10. Fan
11. Case for hardware
12. SD card
13. SD card holder
14. External power supply

Flowchart

AI algorithm



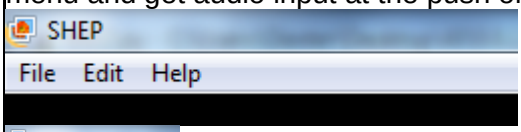
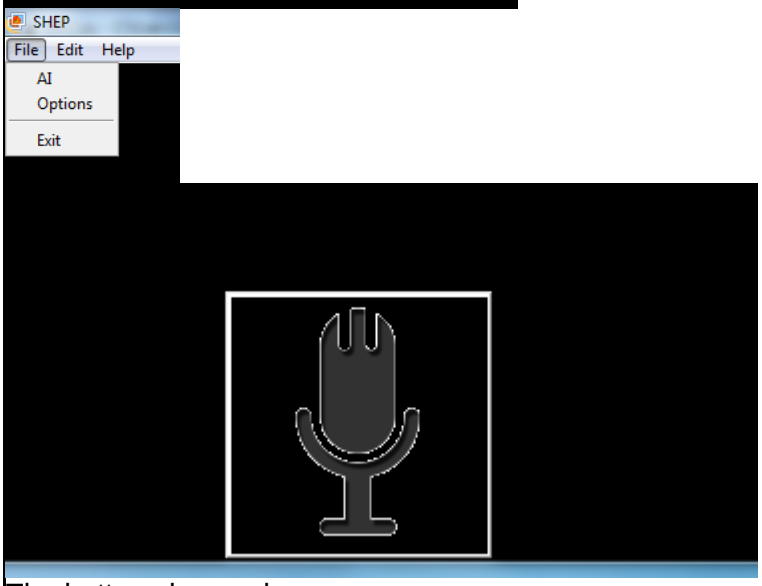
User interface algorithm

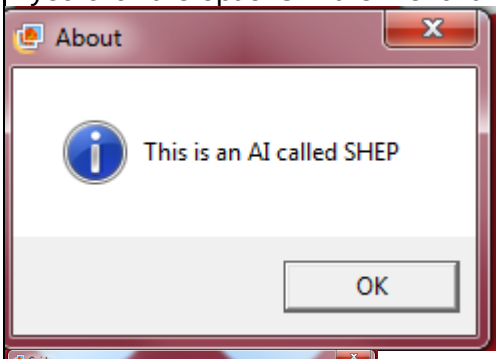
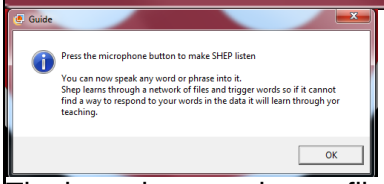
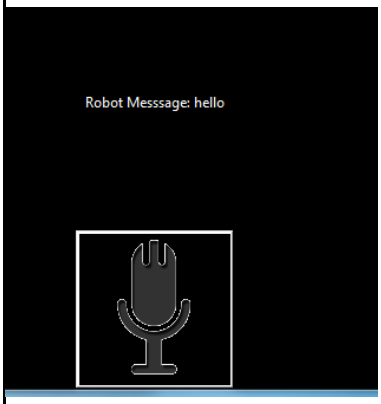
Pseudocode

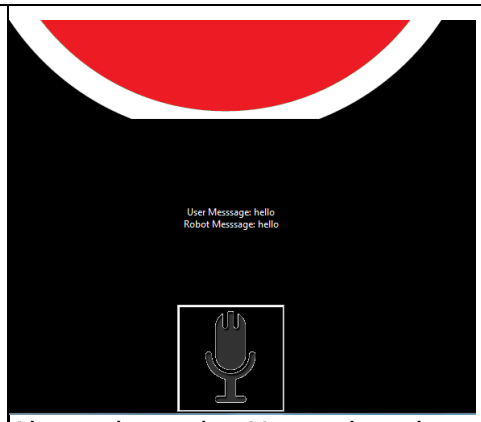
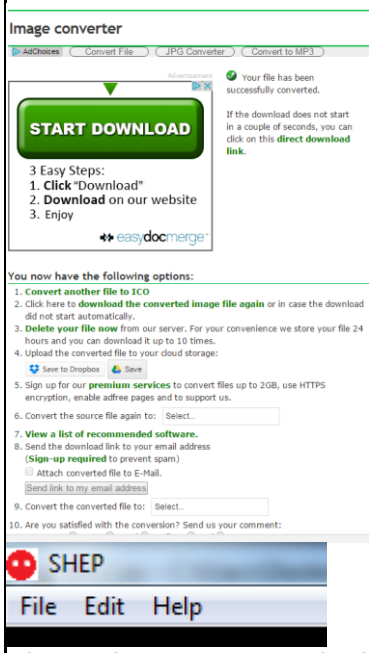
Coded Solution

Development

Development diary

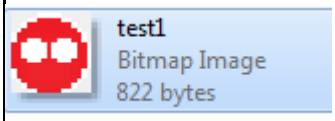
Date	What I worked on	What happened / What testing
09/01/2017	I started working on user interface code. I managed to display an image, a menu and get audio input at the push of a button.   The button shown above. I decided I no longer liked the face of the old version so I redesigned it.  Above shows my new face.	

	 Above shows my application. If you click the options in the menu it would come up with bars with information.   The icon shows an image file which I am not too happy about. I will have to get to designing .icon files. When you press the microphone it will let you speak and then output it on the screen. If the microphone does not understand it simply gives a pop up window.  	
10/01/2017	Today I added in the main AI algorithm. This took some thinking as I cannot use the GUI while my code is running (too much processing for one bit of code).	

	 Above shows the AI search and output part. I started writing this development paper today. I could use some of my plans from before as it is very similar however I had to edit it to make it more relevant to this system. I finally changed the app icon for this. I went and converted my bmp file to an icon file.  Above shows my app. The icon doesn't look the greatest but for now I am happy with it. In the future, I may change this. The system has the functions to learn although I came across a bug. This bug prevents the system from going through the "check trigger" function. I am looking into why this does not work.	
11/01/2017	I got past some of the errors through trial and error using the IDLE shell. I have got to the point it searches the files for data. I need to develop it saving the trigger and learning.	

```
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python34\lib\tkinter\__init__.py", line 1533, in __call__
    return self.func(*args)
  File "C:\Users\Dexter\Desktop\AI\0.0.2.py", line 483, in getVoice
    learn(GInt,message,pathway,Umessage)
  File "C:\Users\Dexter\Desktop\AI\0.0.2.py", line 247, in learn
    trigger = Find_Trigger(Umessage)
  File "C:\Users\Dexter\Desktop\AI\0.0.2.py", line 224, in Find_Trigger
    fpos += 1
UnboundLocalError: local variable 'fpos' referenced before assignment

>>> ================================ RESTART ================================
>>>
...
how ru
how
how
how
how
USER MESSAGE:
...
Oops! Didn't catch that
USER MESSAGE:
...
Juan
error, not in phrase
USER MESSAGE:
...
Wan
>
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python34\lib\tkinter\__init__.py", line 1533, in __call__
    return self.func(*args)
  File "C:\Users\Dexter\Desktop\AI\0.0.2.py", line 483, in getVoice
    learn(GInt,message,pathway,Umessage)
  File "C:\Users\Dexter\Desktop\AI\0.0.2.py", line 247, in learn
    trigger = Find_Trigger(Umessage)
  File "C:\Users\Dexter\Desktop\AI\0.0.2.py", line 230, in Find_Trigger
    pathway = audout()
TypeError: audout() missing 1 required positional argument: 'string'
```

Above shows me working through the bugs.
I have not done as much today as I've been caught up in revision. I have started to experiment with new icons, as I want to add in a transparent background. This would look better and blend into the bar.



I will need to convert this to an .ico file for me to test this out.

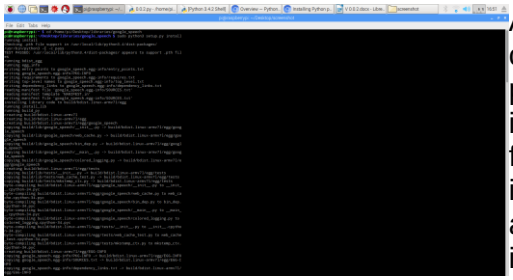
14/01/2017

I am writing this on a raspberry pi!!!!
I have been uploading libraries from my memory stick onto this raspberry pi so I can develop my AI to be mobile.

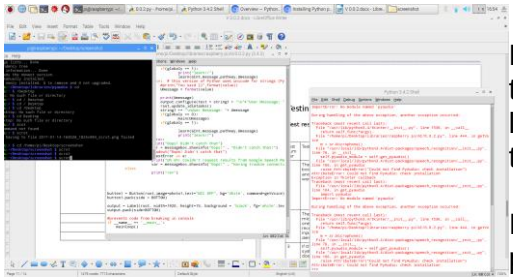
I had to find out how to create screenshots. I managed to do this by researching how to do screenshots and installing software called "scrot" on the terminal. Here Is my screenshot:

Above shows my screenshot.

I organized my libraries and AI code onto the desktop for my own easy access. I had to install all the libraries. I transferred their files over from my desktop and installed them in the terminal.



After running the code and getting past some library installing issues, I found that I must install pyaudio although I already installed it. There was an error in the installing.



Last time I used pip to install some libraries. I went on to research pip.

It turns out that pyaudio isn't the problem. I cannot add a microphone to the audio jack on the raspberry pi as it is only meant for output.

After doing some research into this this problem I came across some solutions. I could buy a sound card adapter for the raspberry pi for it to process sound. The raspberry pi according to element14.com, does not have a line in. the output is based on PWM output and thus the quality is not "perfect".

It has been suggested to get USB audio cards although are rather expensive.

Another option (still pricey) is to add a sound card shield to the raspberry pi instead of use the USB. I like this idea a bit more as I can use my current microphone which is a very good quality.

Once I did further research I found the cheap option of adding sound adapters to the USB port. I can do this for around £6.99. These adapters still use the audio jack which means I can keep my microphone.

I decided it would be a good idea to start off going for the cheaper approach.

I've tested out all the other functions on the software as I cannot currently test out the voice input.

I can work on the learning function of the code by developing the code on my desktop.

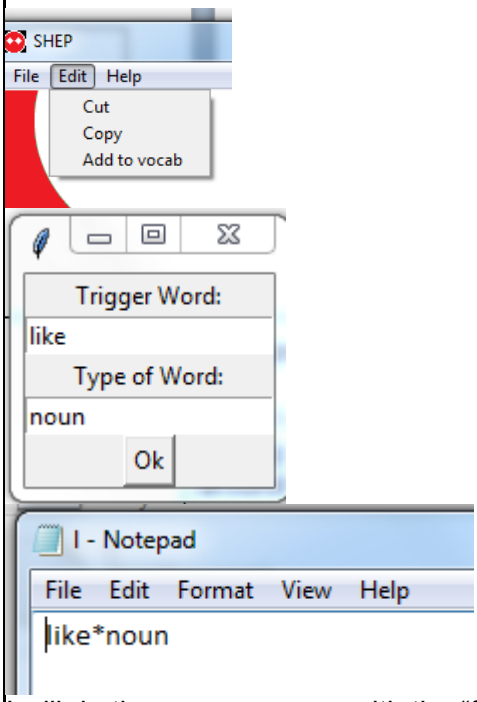
Perhaps I may add different background faces to the AI to show its mood. When the AI doesn't understand something it will pull a confused face. I will develop other faces for it and figure out how I will add them in when the time comes.

My main goal is to get the system learning.

When working through on the computer, I decided it will be easier to add words using the bar above instead of going through a long verbal process. The system will come with lots of words programmed in but if the system does not recognize

words it will tell the user so they can add it another time.

This is the interface:



I will do the same process with the “find subject” principle.
I got to editing backgrounds for the AI. These are the faces:
Tired

Sad

Pleased

Ill

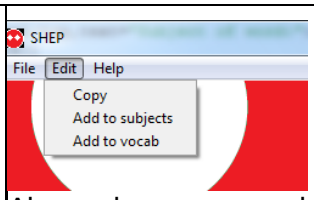
Happy

Fed up

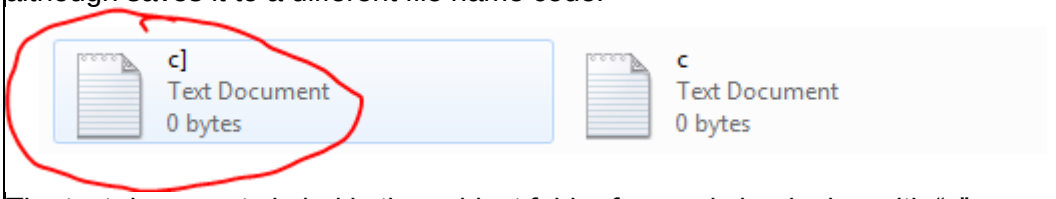
Disgusted

Confused

Angry



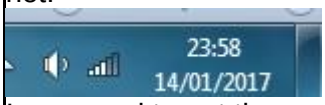
Above shows my new bar. Add to subjects does the same thing as add to vocab although saves it to a different file name code.



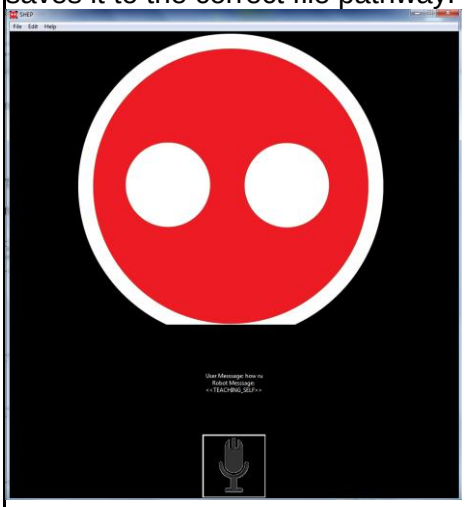
The text document circled is the subject folder for words beginning with “c”.
The normal letter “c” file is for trigger words.

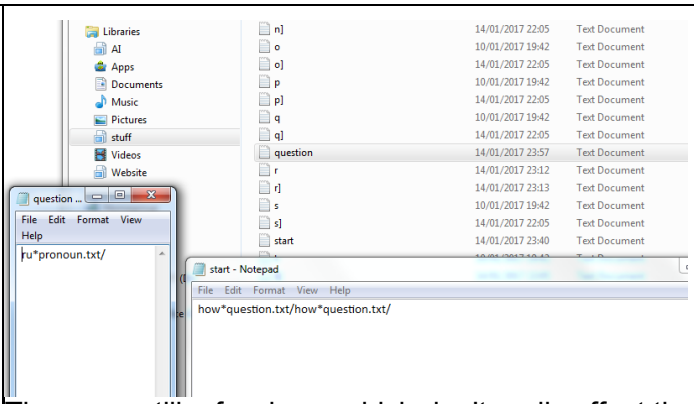
After working on the learning function I have rearranged the algorithm so it is constantly learning (to an extent). The AI enters the learn mode function where it uses its saved vocabulary to fill in any unknown details and will output its response in the data or learn if it isn’t in there. This is all being done in one function.

Currently the AI is having trouble finding subject words.
I was aiming to get past this by 12 as I would go to sleep then if it was fixed or not.



I managed to get the code to a point where it teaches itself all the words and asks you for your input. I must develop it so that it takes in a new input and saves it to the correct file pathway.





There are still a few bugs which don't really affect the way the program runs, just are unnecessary to have.

```
>>>
...
how ru
<e>
h.txt
h
ho
how

question
h].txt
r].txt
loop
r
loop
ru
loop
ru
ru
found

question.txt
file!
0
nothing in file
how ru
```

The IDLE shell is now acting as a debugger.

15/01/2017	After working through hundreds of bugs I was getting to a point in the code where it worked fully. Suddenly my microphone wasn't giving input and I couldn't understand why? I went online to the google speech demo and that didn't work either? This means that my code is fine. No speech was detected. You may need to adjust your **microphone settings**. Above shows the website output. Below shows my shell output:	

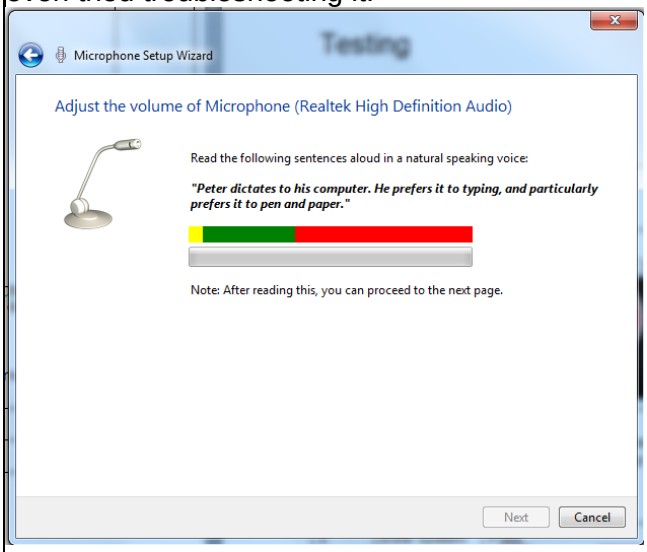
```
>>>
>>>
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\Dexter\Desktop\AI\V0.0.2\0.0.2.py", line 653, in getVoice
    with m as source: audio = r.listen(source)
  File "C:\Python34\lib\site-packages\speech_recognition\__init__.py", line 558,
 in listen
    buffer = source.stream.read(source.CHUNK)
  File "C:\Python34\lib\site-packages\speech_recognition\__init__.py", line 157,
 in read
    return self.pyaudio_stream.read(size, exception_on_overflow=False)
  File "C:\Python34\lib\site-packages\pyaudio.py", line 608, in read
    return pa.read_stream(self._stream, num_frames, exception_on_overflow)
OSError: [Errno -9999] Unanticipated host error

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Python34\lib\tkinter\__init__.py", line 1533, in __call__
    return self.func(*args)
  File "C:\Users\Dexter\Desktop\AI\V0.0.2\0.0.2.py", line 653, in getVoice
    with m as source: audio = r.listen(source)
  File "C:\Python34\lib\site-packages\speech_recognition\__init__.py", line 147,
 in __exit__
    self.stream.close()
  File "C:\Python34\lib\site-packages\speech_recognition\__init__.py", line 162,
 in close
    if not self.pyaudio_stream.is_stopped():
  File "C:\Python34\lib\site-packages\pyaudio.py", line 543, in is_stopped
    return pa.is_stream_stopped(self._stream)
OSError: [Errno -9988] Stream closed
```

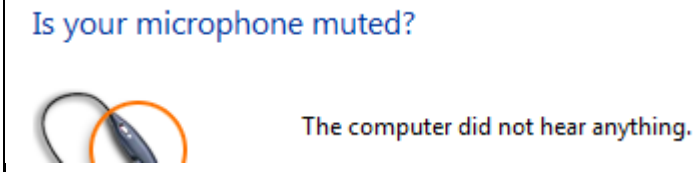
Ln: 6 Col

I decided that I would add another vocabulary data file but this time calls it command. This means it is easier for the program to map out its data.

Trigger → subject → command → Shep's output

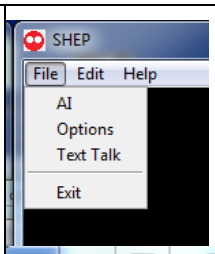
I tested the computer with other microphones and still got the same problem. I even tried troubleshooting it.

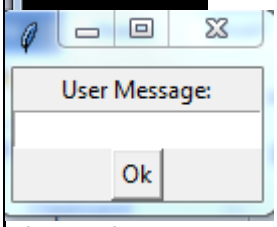




I restarted the computer and still my microphone doesn't work. I will return to this problem a later time. For now I will develop a text talk option in the menu for those without microphones.

If I do this then I can continue to work through the algorithm.



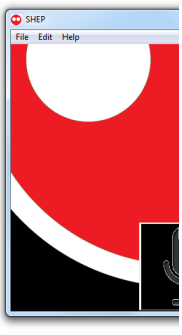


Above shows my text input interface.
My code was getting stuck a lot; this was because my loops were not working properly.

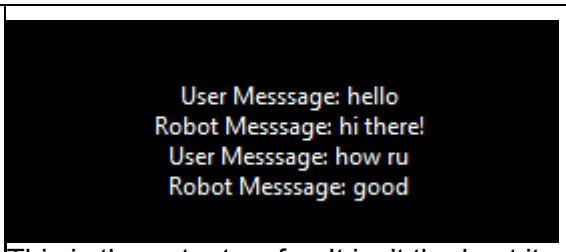
```
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
hello
how
```

After hours of debugging, I finally got to an output point:

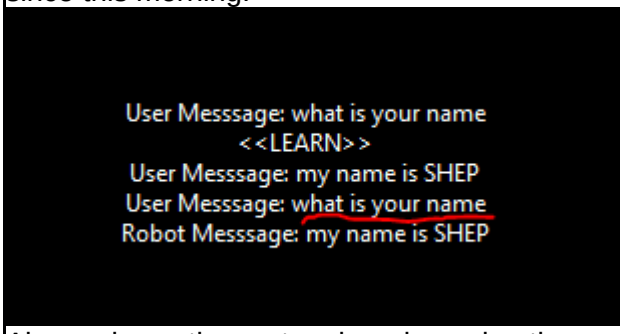
```
>>>
vocab
how ru
h.txt;;
found
w
question
found
1
h].txt
r].txt
found
pronoun
question.txt
question.txt<<
file!
16
ru
how
ru
ru
found
got here
[pronoun.
ru
how
ru
ru
found
ROBOT MESSAGE: good
how ru
```



Now I've got that working I will add in the third "command" words.



This is the output so far. It isn't the best it can be but it has come a long way since this morning.



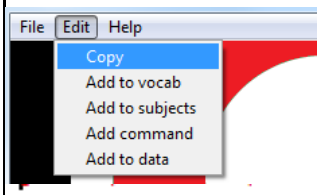
Above shows the system learning using the command words.
"what" = question
"your" = pronoun
"name" = Shep

In the image above I underlined the test to see if it worked.

I also got the copy function working.

User Message: hello
Robot Message: hi there!

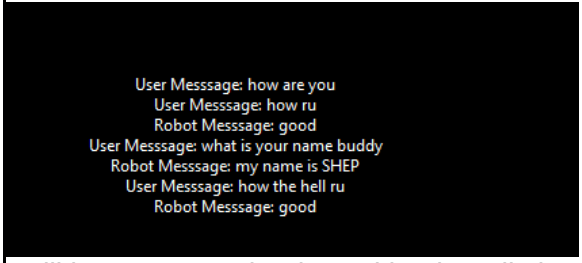
I just pasted the above test in after clicking copy.



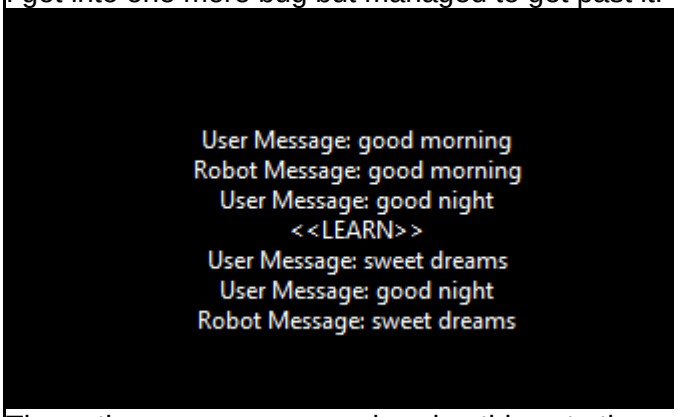

```
root.clipboard_append(string1)
```

Above shows the option and the code which runs.

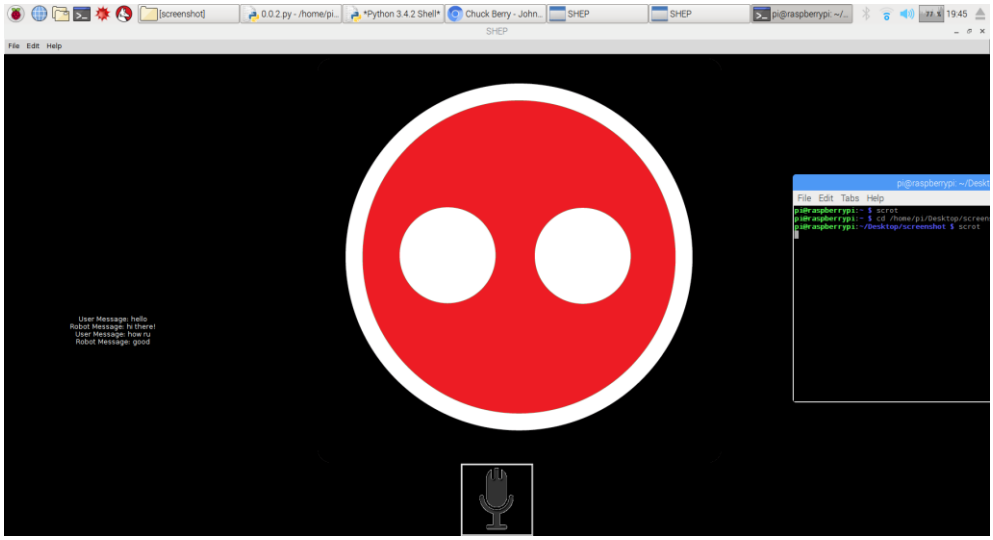
I made sure I experimented with the network of files to make sure keywords worked.



I will have to spend a day writing in a dictionary to the data files. This will be a long horrific process but in the end it will hopefully be worth it.

	It appears I have developed a bug in the code, while everything else works, the learning decided to be either true or false is false when it should be true. I set it to true and there be no reason it should become false.
	When I get to the end of the main algorithm I told it to output the learn variable.
	`>1<`
	`True`
	It is true at the moment but then becomes false when it hits the button decider. It turns out my reset function was being triggered somewhere. I fixed this by putting hashtags behind the functions I need.
	
	I am currently having trouble getting out of a loop again.
	
	I got past that problem but now, even though the new learned word is saved in the correct format in the data, it doesn't work.
	
	The other commands work but this command does not.
	
	I have fixed all the bugs I can see. The data file must be well organized as words added wrong or in the wrong place can lead the AI down the wrong train of thought. Hopefully it can learn ways around this. I will look into this and experiment so the AI can be safe from "brain" disorders.
	
	I worked out a way of changing the background however it froze the main code going around. I will look more into this in the near future. It would be good if the interface could show some personality to increase user experience.
	
	I got into one more bug but managed to get past it:
	 
	
	The pathway was wrong and saving things to the wrong area.	

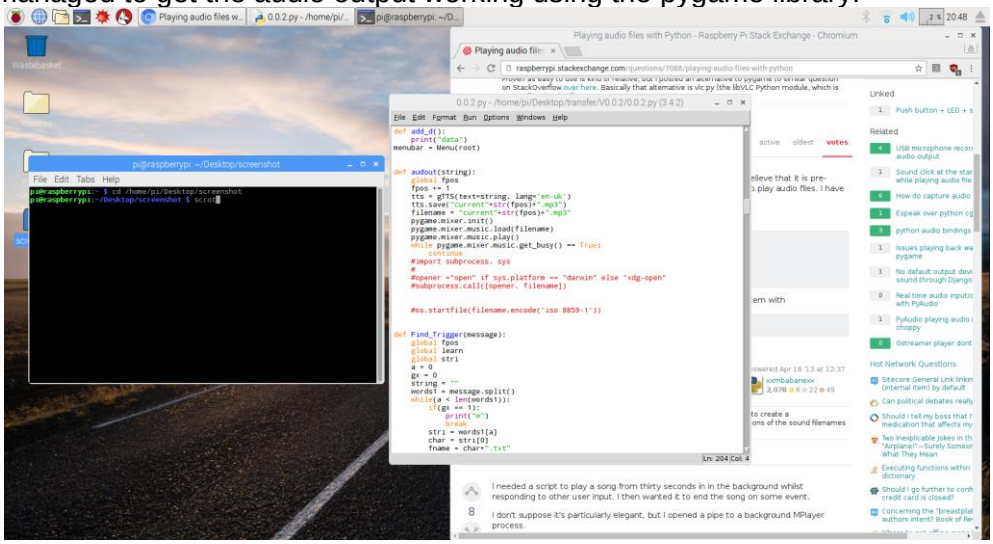
17/01/17	Today I transferred my code to the raspberry pi. The code needed a bit of tweaking but it worked. The only drawbacks were that I need to find out how to play mp3. I cannot use the "os.startfile (filename)" function. I commented the audio output out just for testing purposes.	



The above shows my output. I have not yet received my sound card so I cannot test out the voice input however I have seen people using the same library on the raspberry pi so I feel safe at the moment.

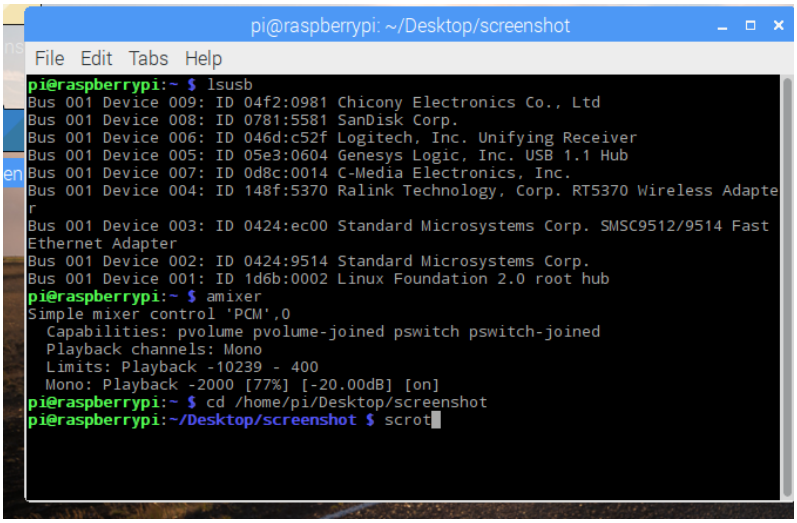
I will have to research further into audio output so my AI can speak. I know that the Raspberry Pi can output sound reasonably well as I've been listening to YouTube on it.

I managed to get the audio output working using the pygame library.



Above shows he part of my code which outputs.

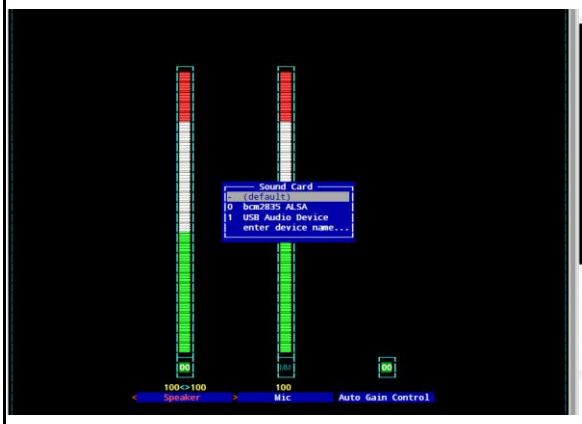
With the AI, I am now left with the physical output. I had a lot of problems with the servos and I had an idea. The user could be able to select what com port to open up for serial communication. This means that the user can add an Arduino via USB. This means I could have a little body with a few actions programmed in and the python code to write to it.
Because I have had trouble with servos previously, I decided that I may use a different form of output such as DC motors. I cannot exactly output to the motors like I can to the servos so it would make it difficult to train in. my plan here isn't to make a learning body, but to make the AI be mobile.

	Once I have created this mobile system and action learning in the code, I will make series functions where the AI opens up multiple functions in order and sends it to the body. I also need to make it so the AI can change its face based on its mood. Some faces are easy to install in such as the confused face when it needs to learn. Happy and sad faces will be a bit harder to install at this time as there is no measurement or variable deciding its mood, unlike some previous AI systems I have made.	
18/01/17	Today I attached my USB sound card to the PI. Obviously it did not work automatically, I looked into drivers and I found that it did not need any. I researched into attaching sound cards and many websites gave me information. Many problems arose as I continued through.  I firstly opened up all the USB ports to check that the Raspberry pi was reading it. The website told me to update and reboot my Pi so I did so. This took around 15 minutes. 	

This did not change anything however. I found that I had to go into some audio files and edit it so that it had the following code inside:

```
pcm.!default {
        type hw
        card 1
}

ctl.!default {
        type hw
        card 1
}
```

The card was changed to my audio card named card 1. This made me the default audio. When I used alsamixer (an audio mixer module) I could see that the default was my sound card. I played around with the mixer and I could hear feedback in the ear phones which means that it is working. When I tried to play music on YouTube I couldn't hear it which lead me to to think that my audio card isn't fully default.



I tried running some tests in the terminal and they were successful. When I tried playing audio in python it gave me the following error.

	found
	got here
	[greeting.
	hello
	hello
	found
	ROBOT MESSAGE: hi there!
	Exception in Tkinter callback
	Traceback (most recent call last):
	  File "/usr/lib/python3.4/tkinter/__init__.py", line 1536, in __call__
	  return self.func(*args)
	  File "/home/pi/Desktop/transfer/V0.0.2/0.0.2.py", line 790, in cleanup4
	  learns()
	  File "/home/pi/Desktop/transfer/V0.0.2/0.0.2.py", line 706, in learns
	  audout(string)
	  File "/home/pi/Desktop/transfer/V0.0.2/0.0.2.py", line 192, in audout
	  pygame.mixer.init()
	pygame.error: No available audio device
	
	Ln: 6
	This explains why I cannot listen to audio on YouTube. There is no available audio device according to the system which means I have not fully set up my sound card. I have followed the sound card setting up to the exact word of the internet.
	
	Perhaps their tutorial was just how to connect it, but I consider it a waste of money to buy a sound card just to listen to blowing down the earphones and see the device name in the terminal.	

19/01/17	The sound card worked today!! I did reboot the system yesterday multiple times, however I did not reboot it at the end as I thought it wouldn't work. Turns out I should never give in to my thoughts. I was always given the advice to walk away from something when it doesn't work and go back to it later. I always found this highly illogical as the problem will still be there when you leave not allowing you to relax; turns out in some cases it works. I decided to check out the AI code to see if the audio output worked... Unfortunately it did. This may seem confusing but the output is now chickmunkfied for no apparent reason? This is most distasteful. I don't think that the audio configuration could do this as I listened to a YouTube video and it played normally. Somehow the Raspberry Pi made a decision of its own, some may say it's impossible but I witnessed it take my folder on my USB drive with all my papers, 3d printing designed objects, code and so forth became a corrupted file!!!! I searched on the internet how to restore this and it gave me multiple solutions. These 'solutions' were the worst idea since diet water! Yes that's a thing... the solution made the device delete this folder so I could not try out the other solutions as the file was gone! This really angered me as that was all my work. Luckily I saved the AI code onto each device and kept some older backups of the folder on my desktop. You may be wondering how I kept this document. Luckily the raspberry pi saves files in the recovery which meant I could recover this paper. Sadly the same	

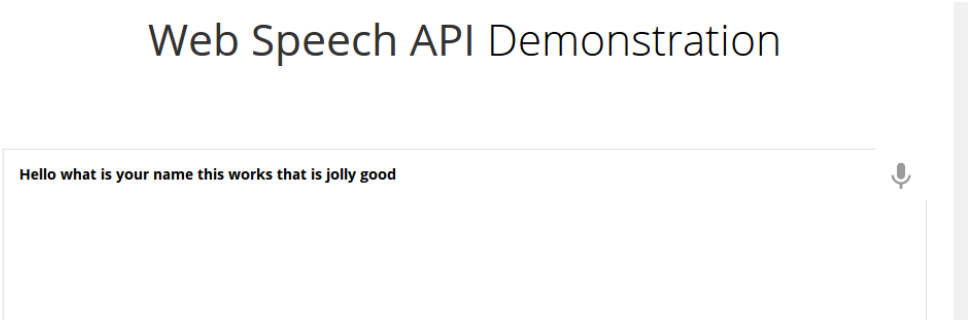
can't be said for V0.0.1 paper. I do have a paper copy of this paper so I could type up the words on it and scan in images (it will lose quality) but this will require a lot of effort.

I do not need that paper at the moment so I do not need to type it up.

This problem has taken up a lot of time and effort today, and thrown a lot of time and effort down the drain from previous days. It is very annoying that this happened.

Usually there is a lesson to learn from events like this, sadly I already backed up files from previous experiences so I can say that I learned nothing!

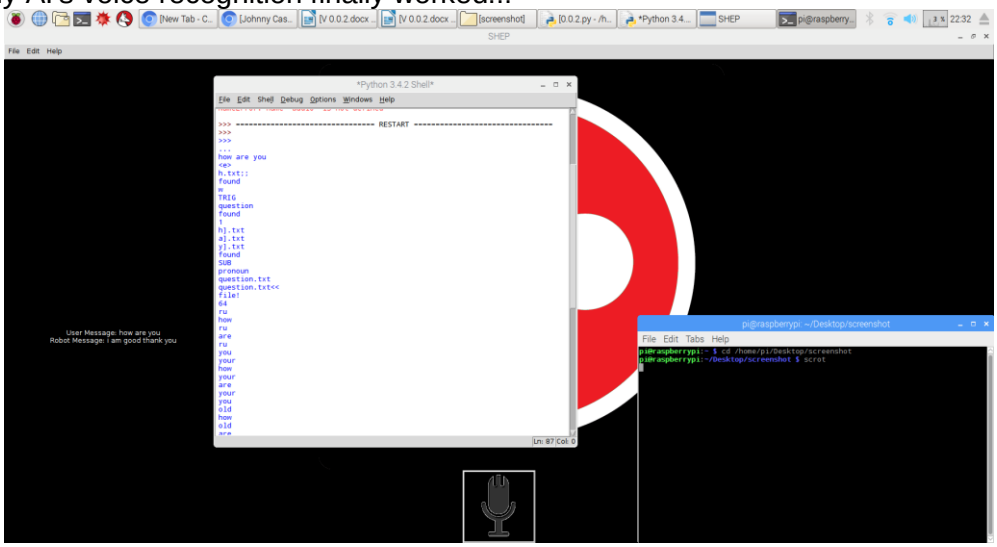
On a more positive note, but not that positive, I managed to get the microphone giving input using a web browser demonstration of the google speech API.

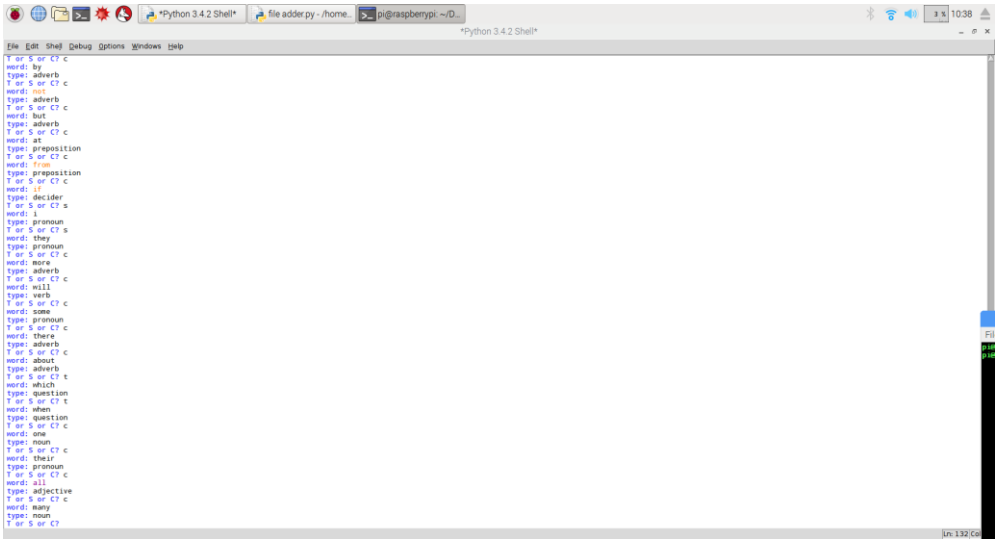


Unfortunately the pyaudio library is having trouble which means I cannot use the microphone with the python code just yet.

```
ython 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
ype "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>

raceback (most recent call last):
  File "/home/pi/Desktop/libraries/speech_recognition/examples/microphone_
ition.py", line 9, in <module>
    with sr.Microphone() as source:
  File "/usr/local/lib/python2.7/dist-packages/speech_recognition/__init__
line 137, in __enter__
    input=True,  # stream is an input stream
  File "/usr/lib/python2.7/dist-packages/pyaudio.py", line 747, in open
    stream = Stream(self, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/pyaudio.py", line 442, in __init_
    self._stream = pa.open(**arguments)
IOError: [Errno Invalid sample rate] -9997
>>>
```

I would research how to change the sample rate but the fact that the code worked online leads me to believe this isn't the problem. I will look at the library and try editing certain factors of it such as the maximum and minimum sample rate.

20/01/17	Today I searched the entire internet for answers to the sample rate problem. This led me across the web on a wild goose chase downloading new voice recognition systems. AT one point I had to reconfigure the default audio device because it changed for some reason? After spending hours searching for an answer I came across a web page with a solution to my original problem! This configures the sample rate within the function. *with sr.Microphone(device_index = 2, sample_rate = 44100, chunk_size = 512) as source: audio = r.listen(source)* The code above is what I found to replace the previous. My AI's voice recognition finally worked!!!  I spoke the words “how are you” and it registered “how are you”. On the desktop it registered “how ru” which is leading me to think that the sound card I installed is pretty good.	
21/01/17	Today I started adding words to the database. I don’t want to add too many at the moment but I would like to add the fundamentals. I wrote some separate code to make it easier to add words to the files instead of the GUI way. I searched up most used words in the English language and it gave me 2000 words. Lots of them were connectives. I added around 43 words to the current AI data.	



I don't want to add too many words at this point as users may not even use them. The user can add words as they go along.

I also fixed another bud within the system as the code wasn't adding unknown words to the trigger folder. It now works and does add data.

When using the AI I thought to myself, what if I want to change the AI's response? This gave me the idea that the system should have a DBMS (database management system) so the user can edit the AI's responses.

In the previous paper, I set myself 6 goals for the next version.
I have completed and haven't completed:

task	completed
Better mobility using raspberry pi	yes
Use login for AI and access to the WIFI	no
A database of words and their type so the AI can teach itself trigger words	yes
Add a series of functions when teaching an action	no
Use a bootloader on a file to save user specifications	no
The GUI is improved not to freeze and to be accessible	yes

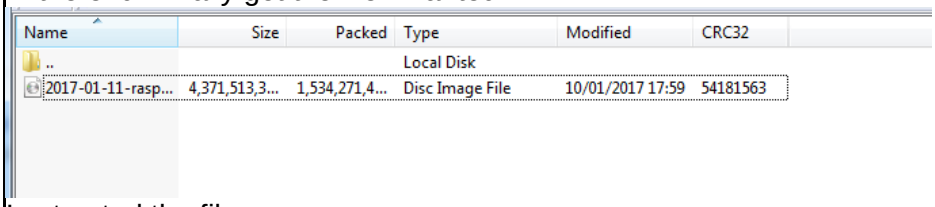
I have completed half of the tasks so far. I next need to add the action functions. For this I am thinking of just adding it in the menu to add a function, then you call upon this when teaching the system. I spoke about using just a little robot chassis instead of servos as the servos was proving to be too much of a problem for now. I will program an Arduino to control DC motors for now and pre-program in functions. The AI will create new functions every time as if you did have a servo board in but lots of functions will be ignored by this board.

	I tried again this evening to slow down the output speed of the AI. I still have got nowhere with it! I was looking into sample rates, people on the internet have played the file at different speeds however when I tried it didn't work. I will have to research into this as the high voice is annoying and hard to understand. This makes the program less user friendly.	
22/01/17	Today I was trying to add a touchscreen to the raspberry pi to make it more mobile. Unfortunately it was slow and pixelated. I looked online and it appears I need to install drivers. These drivers came as a .tar file. I had software which I used to use to extract these files. It appears it was only a free trial. I cannot use this software anymore unless I pay £16 minimum. I shall not give into these capitalists trying to bankrupt me on unneeded software! The SD would not let me write this unzipped archive folder to it as it would corrupt the SD and I had to format it each time. This is still a problem. I plugged the HDMI on the AI onto the TV screen today. This allowed me to conduct a beta test on volunteers. I found that the algorithm works fine with the AI, it is just the management of subject and command words. My findings show that you must be specific to get the answer you want, this means you cannot use connectives or anything of that type as subjects or commands. If you do you may end up resulting in the AI answering outputs to different inputs. This gets very confusing to explain. I will need to give more direction to the user on how to use this software. This is where my bootstrap loader comes in. next time I will create the bootstrap loader which saves specifications of the user, this will also give a guide and perhaps interactive tutorials on how to use SHEP. This prevents the user from breaking SHEP. I could add a reset to factory settings button if the user fully messes up the system. Hopefully, in the next version the user can use a DBMS to make sure everything is perfect. Today I looked into motor control; I have done this before but for some reason I could not master it today. This then led me back to look at servos again. I may try out the servo input method first with the AI as this is how I want the AI to learn. Today was more of a research day, a non-official test day and an administration day. Today I've planned out what I'm going to do over the next week. This is: • Make bootstrap loader • Start-up tutorial • Servo learning • Series of action maker	
23/01/17	Today I was looking into extracting files and uploading operating systems to my	

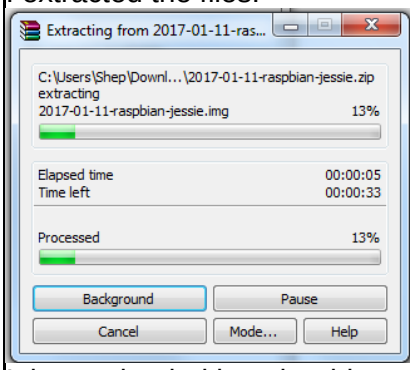
SD.
I was mostly unsuccessful before I realised I was using the wrong type of OS, I needed to use Jessie for the newer boards. The reason I am learning about extracting and writing to the SD is because I want to build up to writing the touchscreen drivers to the SD.

The Jessie file takes ages to download. This means I have to stay up late to get the file and write it to the SD. I prefer doing the programming but unfortunately I have to do my own hardware as I do not have a team of engineers working under my command.

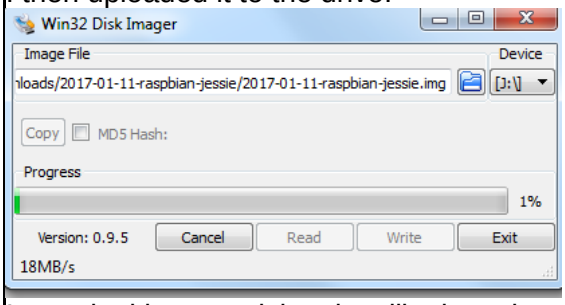
In the end I finally got the file I wanted.



I extracted the files.



I then uploaded it to the drive.

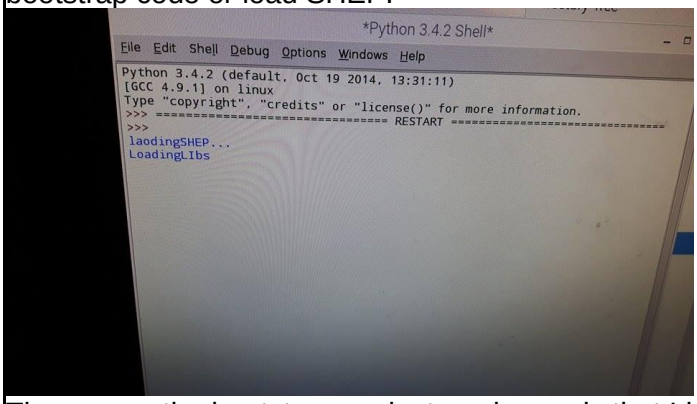
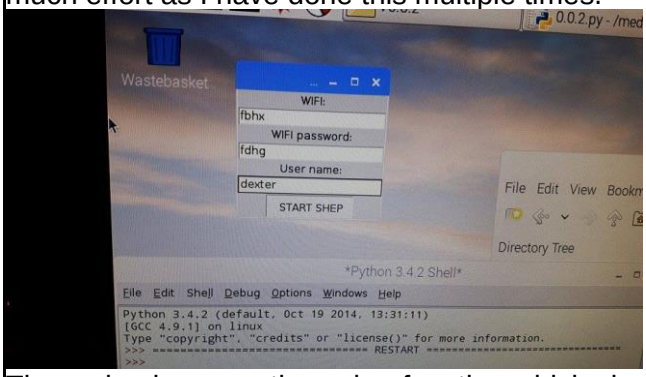
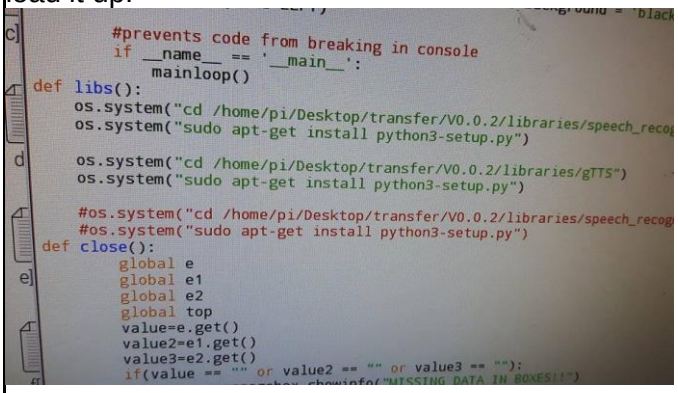


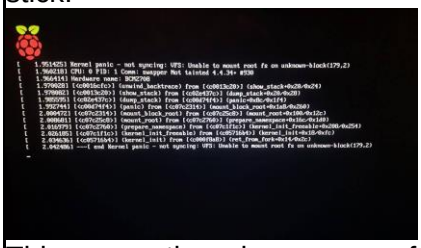
It was looking promising, just like how the video looked.
On another note, I was reviewing the touchscreen file and I could not find an image file to write to the SD card.

I ejected my SD after installation and my raspberry pi booted up using the Jessie Raspbian OS.

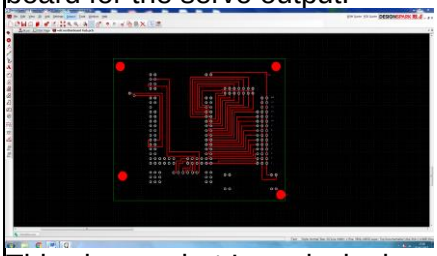
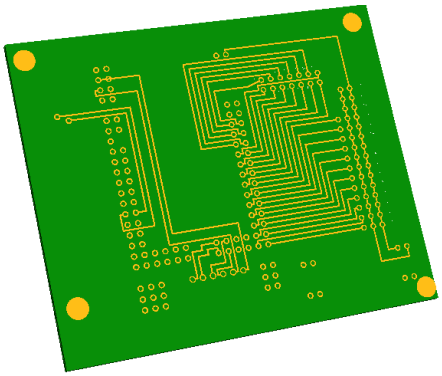
Now I've done this I can research into the touchscreen API/Driver and try get an image file I can upload.

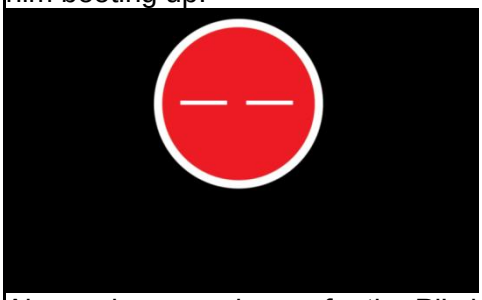
If cloning an SD may not work, I will have to develop the AI on memory sticks or external hard drives for the raspberry pi. There is still the problem of installing libraries. I understand that using the OS library I can send commands straight to the terminal. I could have a setup piece of code which boots up a raspberry pi to run SHEP. I could incorporate this into my bootstrap loader.

24/01/17	Today I worked on the bootstrap loader, I got it working to a degree. I have made it so you add in your Wi-Fi name, password and user name (the only options I can think of to add right now). I got the code to enter this data into a text file. The code checks if there is anything in the file to whether it will do the bootstrap code or load SHEP.  The reason the bootstrap works to a degree is that I have not programmed it to actually connect to the WIFI using python as I have not found out how to do that yet. The user interface of the bootstrap I feel I must improve; this won't take much effort as I have done this multiple times.  The code also goes through a function which should download libraries needed from the internet using pip. I am using the OS library to write directly to the terminal. I have tested the individual commands; I just need to test my AI out on an empty raspberry pi (which I have). I shall move my AI to a memory stick and load it up.  	
25/01/17	I had trouble installing libraries on the raspberry pi again. I managed to do it before but now everything I try does not work. It is to my understanding I can clone the SD on the Pi. I will wait until I can buy a memory stick with larger than 8GB to transfer data. I am currently using all my memory sticks so I will have to	

	buy a new one. I briefly tried to do this today but there was a size problem with the memory stick.  This means there is no reason for the OS library download as all the raspberry Pis will be configured to run this code already.	
26/01/17	I have attempted to clone the SD. I was able to clone it to a memory stich but I could not write the content to an SD for the raspberry pi. This gave me the error of: “waiting for SD” I need a micro SD to USB converter so I can write the cloned OS directly to the SD. One thing I did get working was my screen today. The touchscreen does not work because I did not install any drivers. The output however does work as I went into the operating system and changed the HDMI output to a smaller range. What I shall now do is add a fan into the back of the raspberry pi as it is getting rather hot and over the weekend 3D print a case around SHEP's head. All I have to do now to complete this version is to make the Arduino action learning functions and WIFI Adding ability.	
28/01/17	Today I worked on the action function of the code. I wasn’t too fussed on the Arduino code, I focused mostly on the python algorithm. I made a window you open up in “edit” which allows you to enter your sentence. This works a bit like the type option, it decides if it is already a command or if it is not. If the command is not, it sends the transmission to the Arduino and saves an action code in the data. Later on when I said the command I typed in to add as an action, it sent the appropriate transmission to the Arduino. This proves that the action code on python works. I have not tested this out with the servo board yet, neither have I added the series action command. I only had a short amount of time today to work on the AI as I have to fit in revision for exams; hence I only got one thing done today. After thinking today, I will need to pre-program in the command “stop” to transmit data to the Arduino to reset it, this will be useful when adding commands. I will also add a 'turn off learn' function in the menu if you accidentally said something wrong and the AI wants to learn it. This can cancel you adding	

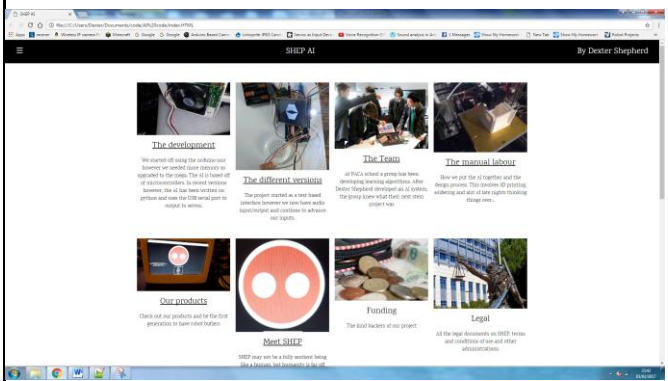
	anything wrong o the AI's brain. I added more ideas to the “future ideas” section of this paper. This also led me to write more plans for the next version and what needs to be added. I will however finish this version first before any of this theory becomes practice. Today I’ve had a case for the raspberry pi and the touchscreen. I’ve been thinking about the servo input and output. I will test out the servo output tomorrow. From my memory of version 0.0.1, the Arduino + python code worked fine, it was the servos drawing too much current from the servos. This can be solved using more power on the servos and fans on hot places. After I manage to get the servo input/output working, I will design a PCB mother-control-board for the Arduino and fans. This board will need: • an SD card interface • LEDs for debugging • computer fan plug-ins • power input • Arduino controller	
30/01/17	Overnight I cloned my SD and it worked! I loaded up my code onto another device and all the libraries were working! I decided to spend the day revising... with SHEP. I made SHEP learn some facts about all the subjects I am doing GCSE. SHEP learned a lot of facts. While going through this, I decided on a superior way to organize SHEP's data banks. The subject word should be a file individually. “Hitler” leads to person file. So if I ask “who was Hitler” it tells me “leader of Nazi Germany”. I told it to do this. The problem is if I ask “when did Hitler become chancellor”, this gives me the same output. Hitler should be its own file with a who, what, when, where and why approach. The user will only add the name of the subject, no longer what type it is. I changed this so that this would work, but there are lots of files currently which used the old system s. I cannot add much more information about Hitler for example. I worked on start-up code today, after testing out my python script in the TXterminal, I got SHEP working. I figured it would work in the same way if I booted it up. Unfortunately not. SHEPs GUI booted up with a black square on the right hand side. There was also no mouse cursor so I couldn’t find my way around. If I got the touchscreen working this wouldn’t be a problem, I could also find ways to work around the giant black square. I came across an online tutorial on how to get a raspberry pi working with a touchscreen. https://www.raspberrypi.org/forums/viewtopic.php?f=45&t=108432 https://learn.adafruit.com/adafruit-pitft-3-dot-5-touch-screen-for-raspberry-pi/easy-install . The idea of downloading new image files for the OS worried me. I did not want to reset my OS, if that is what I was doing. I decided that I will back up my raspberry pi SD before I do this.	

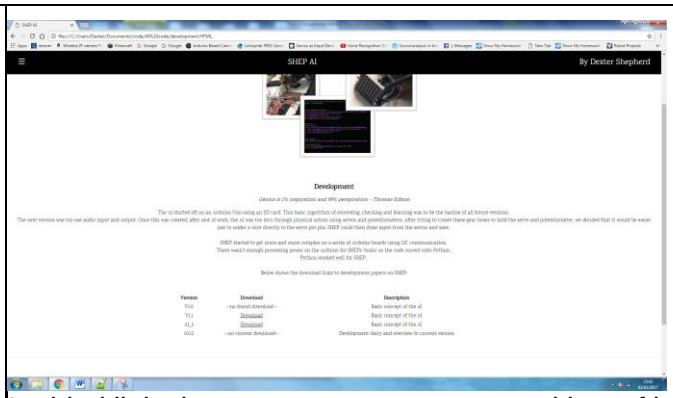
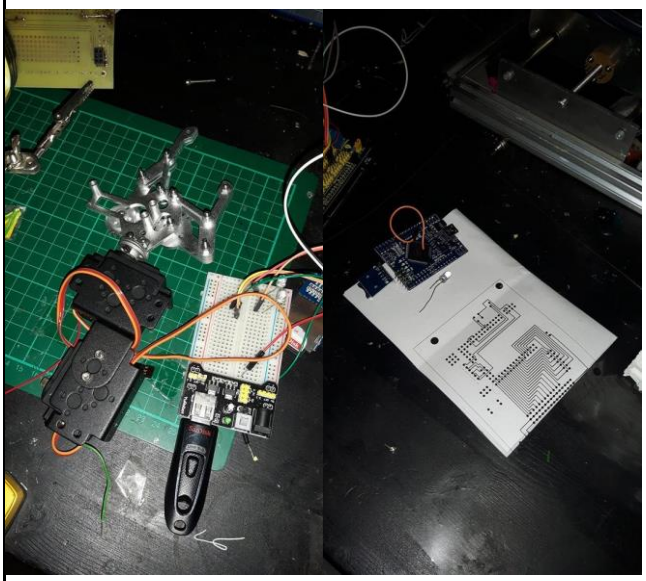
01/02/17	Today I fixed the problem of SHEP booting up by asking my computing teacher. It turned out I had to add a “&” to the end of my booting message so the system could load up the main OS before the python code.	
02/02/17	I would like to test the action input however, I cannot find a spare SD card at the moment tot write data to on the servo board. I am thinking about making a PCB board for the servo output.  This shows what I am designing which shall later be a PCB. Below shows a 3d view he software made:  Once I can train a few servos (only a few for now as there is potentially a current problem still) and it works, I can develop series commands. Once I’ve developed series commands I have finished this version. At the moment I’m quite drawn between SHEP’s applications; I would like him to be able to be a pliable, as I've mentioned before. However, if I give him the ability to output through servos that will be more than useless in a system which is connected to a server! Possibly I will have to add certain options which disable actions when connected to a server, or the actions just go through to an Arduino board where it is defused and the data is ignored. This would be beneficial as the system does not have to output. The AI in its current form is more of a mobile home robot to do the daily chores and so forth, all though I see that the as potential is far greater. I would like it so SHEP can be plugged into a network. This network can be a local area network or wide area network. Doesn’t really matter as long as it can access the data on the server. I’m very happy with how SHEPs turning out right now compared to previous system on Arduino. The Raspberry Pi has higher RAM and processing power than the Arduino, this makes Raspberry Pi voice recognition and Internet capabilities more effective than the Arduino. I’ve already planned points for the next system which is more of these algorithms to replying to messages I’d like the AI to be able to learn logic and really think about what he is replying to the user. After I have improved the user interface algorithm, I want to start	

	developing SHEP to have server capabilities meaning he can plug into the system using most likely Ethernet. SHEP can then access data on the system, which should be easy enough using the terminal and OS library. Then SHEP will be able to access the system data up to and including CCTV. This means that SHEP could access video from the CCTV and process it on computer devices for facial recognition. This further means that SHEP could control, maintain and look after buildings, and furthermore cities. Back to SHEP currently, I will need t acquire an SD card son to test out servo input with the AI. If that works I can develop the PCB over the weekend, and in the meantime work on the series of action learning. The background of the raspberry pi is the default OS picture of a road. It is a nice picture however it boots up with this flashing on screen before SHEP fully loads. I was thinking about having a sleeping SHEP face on the screen to show him booting up.  Above shows my image for the Pi's home screen. It was formatted to the 800x480p screen resolution. I installed this as the background.	
03/02/17	Today I didn't work on SHEP much however I did work on advertisement. When I've finished SHEP fully I will need a way to get him out there. Today I made a website for the future of SHEP. This was reasonably easy using HTML code.	

```
<!-- Sidenav (hidden by default) -->
<nav class="w3-sidenav w3-card-2 w3-top w3-xlarge w3-animate-left" style="display:none;z-inc
 <a href="javascript:void(0)" onclick="w3_close()"
 class="w3-closenav">Close Menu</a>
 <a href="index.HTML" onclick="w3.close()">Home</a>
 <a href="#about" onclick="w3_close()">SHEP</a>
</nav>

<!-- Top menu -->
<div class="w3-top">
 <div class="w3-black w3-xlarge w3-padding-xlarge" style="max-width:40000px;margin:auto">
   <div class="w3-opennav w3-left w3-hover-text-grey" onclick="w3_open()">☰</div>
   <div class="w3-right">By Dexter Shepherd</div>
   <div class="w3-center">SHEP AI</div>
 </div>
</div>

 <!-- Pagination -->


 <hr id="about">

 <!-- About Section -->
 <div class="w3-container w3-padding-32 w3-center">
   <h3>SHEP</h3><br>
   <img src="assets/pleased.png" alt="FINISHED_PROJECT picture" class="w3-image" style="dis
   <div class="w3-padding-32">
     <h4><b>SHEP</b></h4>
     <h6><i>A computer would only deserve to be called intelligent if it could deceive a hu
     <p>The system is able to read a main file and create files and create pathway links de

he system has speech to text for its main input.<br /> By using serial communication, the A
br />
he AI has audio output text to speech on top to make it a full audio I/O system. The systen
br />
he system has a data file organized alphabetically with every word in the English language,
br />
his system runs on raspberry pi and desktop although SHEP is programmed into the Pi.

br /><br />Lots of people ask; "why is SHEP called SHEP?". SHEP is named SHEP as its develo
br />and his name could be shortned down. Dexter named his AI after himself because it buil
br />Dexter Shepherd feels that SHEP being named after him shows more passion and commitmen
</div>
 </div>
 <hr>
```

I enjoy making website applications as it is creative and logical at the same time.



	 I added links between pages, a menu, and lots of info and downloads to old code and development papers. I have nearly finished this website so give me a few more hours and I am ready to host it when it needs to be online. I will need to get an SD card to continue with SHEP right now.	
08/02/17	Over the past few days I trained the system to output through a servo, after i trained in the servo. I need to run a few more tests to make sure the servo is accurate, but hopefully I will be able to develop the series function soon. I am developing a PCB to drive servos; I am also re designing the body so I can fit all circuits in and not have wires everywhere. The AI must be neat. I am close to finishing SHEP. I have been cloning the OS with SHEP on recently, this is so SHEP can be backed up, and I can edit an auto Shep mode which requires no GUI and is more pliable to work.  When writing a series command, I think the python code should gather an array of commands after following each pathway, after this the array will be sent to the Arduino where it outputs the command.	
11/02/17	Today I attempted to add the Wi-Fi function to the python code. After many failed attempts I started thinking; ‘Why add a function which is already on the Pi’. The user can use the raspberry pi as it is and have the SHEP software installed. To protect SHEP I will need to eventually convert the python code to exe format.	

The code could still obviously be reverse engineered from the machine code; the SD could also be cloned.

I will need to copy right or patent the SHEP software to make sure that people buy from me.

I like the idea of open source; however I like the idea of owning a yacht!

Above my dreams is SHEPs dreams and development, SHEP will need teams of engineers to develop him further than my abilities. These engineers will need pay so SHEP will need to make money. I am planning to sell SHEP (when the software is in top condition) in many forms. These consist of:

- Robot
- SD card for existing Pi
- Server SHEP
- Education buildable robots

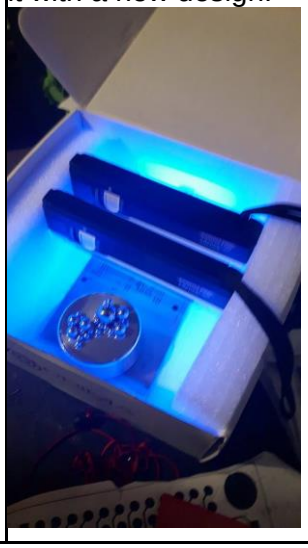
I predict SHEP will be functional for these tasks by Version 0.0.5.

Now I've planned out SHEPs technological functionality, I can draw up that all I need to do now is develop SHEP to learn series commands. I need to finish the PCB board for servo connections, then do further testing on the accuracy of output. Once this is done I can develop the series command.

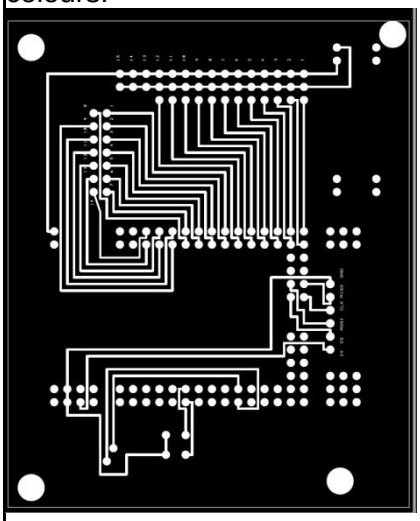
I have already started writing the series ability in my Arduino code.

I am continuing to develop SHEP to work automatically and listen out for keywords or phrases. The user will have to open SHEP up and select this as an option; the user then can switch of the monitor and have the circuit working by itself. If the user wishes to make SHEP controlled using the GUI again, they plug the monitor back in and de-select auto mode. This function is a side project I hope to develop for the next version. I am starting it now as I have spare time when I am waiting for parts to arrive from amazon.

I made a UV box and tried creating my PCB. This worked to an extent; however there was not enough weight on the PCB so some parts of my design did not transfer over. I realised I have to transfer inverse colours after watching a video; I had to scrape off all of the photoresist paper currently on the PCB and replace it with a new design.



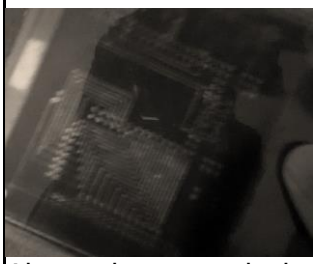
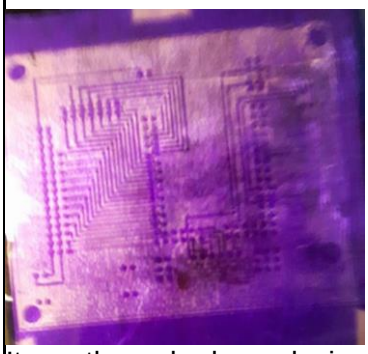
It took a while but in the end I was able to reprint the design with inverted colours.



Today I've continued to 3D print buildings for my model city which I will build a SHEP server for and test out SHEPs ability to control a LAN of cameras. This print went on for a while, is currently 21.46 and I have used the percentage currently and the time taken to calculate how long I have left.
On 41% and a time of 9:25 (hrs:mins):

$$\frac{\left(100\times\left(\frac{(9\times60)+10}{41}\right)\right)-(9\times60)+10}{60}=13.524..$$

I calculated a remaining 13 hours.
While waiting until a convenient time to go to sleep, I started to work on the auto function of SHEP.

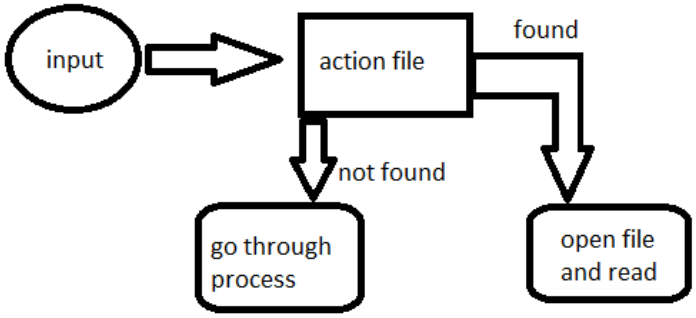
12/02/17	Today I finished off the auto run code. Once selected, the code waits for the trigger word of “robot”, once found the code signals a beep sound. This tells us that you now can say your chosen phrase. For example you could say “how are you”, the AI will reply “I am good thank you” and then make a different beeping sound to show that it has finished. The finished beep signifies it is listening for “robot” again. There are still a few bits I must tweak such as the pop up which appears if the code does not understand. I think this may be slowing down the listening function. This will be investigated further. Today I re-did my PCB, I inverted the design and exposed the PCB to the UV light.  Above shows my design on top of the board. Below shows my PCB after UV exposure.  It mostly worked, my design was transferred. The printer did not work properly as the picture is very ‘dotty’ rather than fully black. This means some of the UV light got through to the PCB and made little dots. As long as they are not too grouped together, the board will not cross circuit and everything will be fine. Today I went up into the loft to look through my old childhood toys. I found lots of toy cars. When SHEP gets camera abilities I shall need train in the cars and add little number plates. These cars will have a good use for the training of SHEP.	

	 Above shows two of the 3D printed city buildings. There are a lot more on their way. While a wait for the chemicals I need in the post to continue with this, I shall do another mini project (like the auto mode). Perhaps I could develop a switch or a memory stick check when the code boots up to decide if it will start on auto mode or not. This would be helpful for robots doing trained tasks not under human supervision. I will also get on with school homework as I have a lot for the holidays.	
14/02/17	Today I finally got round to editing the action adding to two buttons, series and single. They both lead to the same area however I am planning to program the series button to lead to another window which lets you type in commands to add. This will create a long file of bits to send over. After this I shall program in the ability to find these series commands and output them to the Arduino which saves to a list and outputs the files to the servos. The version 0.0.2 is coming together now. I have a clear plan for what i must do, I have plans for extensive testing (further than this papers testing area). The AI action mode must be tested by getting the AI to do tasks such as an arm feeding my goldfish on command and getting an arm to throw a ball. Then get the AI to do both. The AI will be tested with others (beta test) to make sure it is ready for its next step in evolution. I've planned out what if want to do with SHEP for the next version. The next version is very much about improving his thinking. SHEP will be able to reply based on the subject of conversation and remember what the subject is. SHEP will be able to answer long text. "How are you this is nice weather" it in its current form will answer "I am good thank you" where as it should answer in the next version "I am good thank you. The weather is nice" SHEPs version after this will be for developing sensors such as vision.	
16/02/17	Today I worked more on the PCB and series action mode. The PCB has been developed using sodium carbonate and has been put in overnight ferric chloride to etch it.	

The above shows images of my PCB. This is only a prototype as it is the first time I have used the UV light method for PCBs.

Within the code I have added the series button to get input from the user and add it to a string. I next need to develop it so the code can add multiple functions from the user on a loop until told otherwise. It also needs to search a file for the series of functions on each load up (this makes it easier to code but slows down performance time).

The system will work like this:



I need to add this ability next time. In the morning I shall check if my PCB has worked. Then after revision and school work I shall continue with the code.

I was thinking, if this AI works out perfectly, after the next version, it would be great to have this as an inbuilt feature of the raspberry pi. It would be good to work with the raspberry pi foundation. This would require no money and get SHEP out there in the world. This would also give me the opportunity to work with a team of programmers.

Obviously this is a way of dream and I shall have to see what my options are nearer to the time.

10/04/17	Over the previous weeks I have been rethinking my strategy for SHEP. I have	

worked on an Arduino which can enter data into a computer though USB the same way a keyboard does. The Arduino has only been programmed to send through the letter ‘A’ on repeat. I am planning on attaching this board to a series of Arduino boards entering data. This can consist of cameras, accelerometers, temperature sensors ect… I will attach another raspberry pi with voice recognition code running to send through to the main board containing SHEP.

SHEP has been re-done so it only uses text input, this means USB input devices can talk to SHEP. I feel keeping the main SHEP code separate from the system input/output code will be better as I can allow users to modify the inputs and outputs to their specific needs. This makes SHEP partly open source, although the main SHEP software is not.

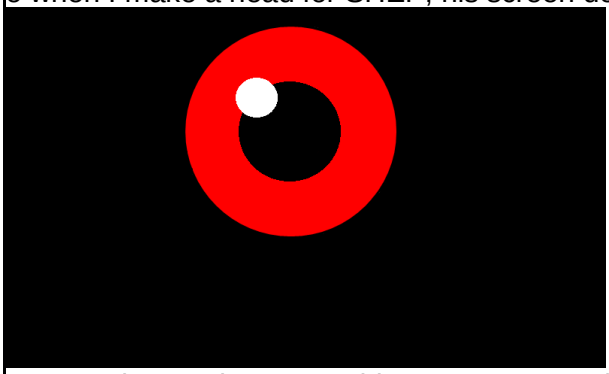
I have been designing the head of SHEP, unlike before, I am designing it to fit everything inside and be simple to print. I am using a design I have used with older AIs; this design is a hypercube with the eye at the front. I have reformed lots of it so the eye fits a screen in, and the back is a ‘criss cross’ frame to fit a fan on and easily screw down the circuits.

The design is only on paper currently, but it shall be designed on a CAD software soon. I am guessing the 3D printing will go on for a matter of days as I am printing a lot. My 3D printer bed is currently broken and not heating up due to certain fire problems.

SHEP will use an SSD to store his data, this will increase his memory. The battery of SHEP will consist of 2 external phone chargers supplying 5.1 V each. In this version, I do not think I will add the action input. Instead, I would like to add the ability for SHEP to learn through imitation. In planet of the apes I remember seeing ‘monkey see monkey do’ when the humans would teach the ape to do something and then get it to do it. Instead of using feedback servos using a lot of power, I could use USB robot arms.

When I was 10 I remember building a robot arm, there was a choice of building a controlled one or a USB one. I chose the control, but looking at it now I could use a USB one to send data through. The input/output pi can be taught to do an action or send through a voice command. Then the human does something for the AI to follow. I shall not be adding this ability in this version of the software.

On smaller news, SHEPs face has changed. Shep is now an eye, this is s
o when I make a head for SHEP, his screen does not display another head.



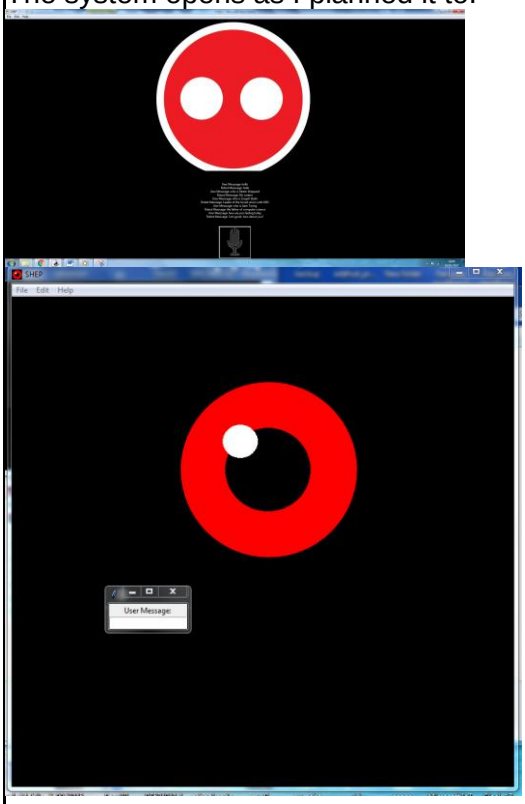
An eye always gives something more personality, rather than the generic computer face.

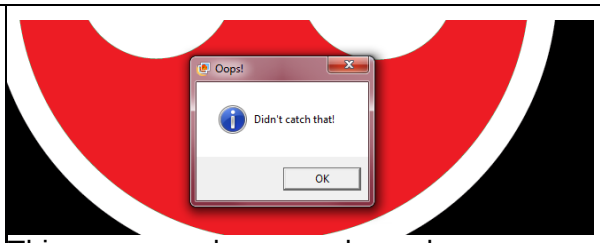
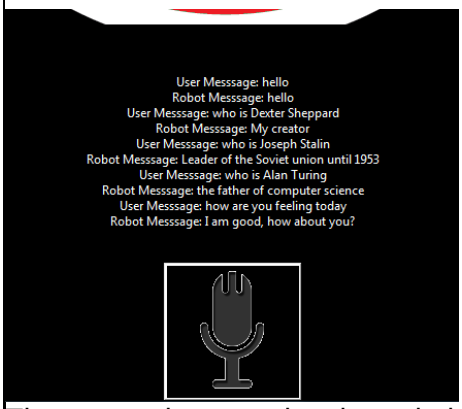
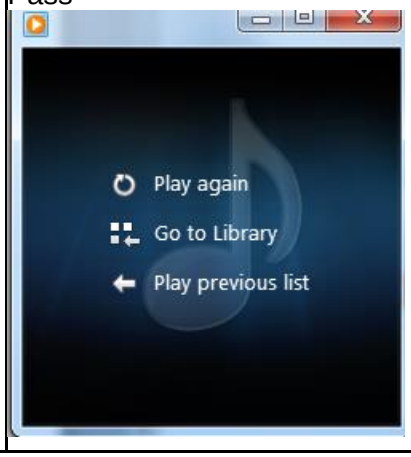
	In future versions I would like to add the ability for the AI to fly around on a quadcopter drone. This is more efficient than robotic legs as it can travel simply over obstacles, it also requires less balancing and there is more support online for building drones.	
12/04/2017	I have been getting rid of bugs within the code and reorganized the code to work in the new way (where the subjects have their own file). To ask “how are you” is now illogical as there is no subject. “how are you feeling” is the correct question. I have been looking into how much the average SHEP model will cost, it is a lot. SHEP requires: • Two raspberry Pis • 2 Arduino boards • Ethernet cable • Fan • Hdmi cable • GPIO cable • 5” screen • Two external phone chargers • USBs • 3d print filament • Text to speech module • WIFI dongle (long range) • USB sound card • Microphone • USB powered speaker • Screws	
29/05/2017	It has been a while since I physically worked on SHEP. I am currently half way through my exam week and haven’t been able to spend the time I wanted to on SHEP. I have decided that to get SHEP out there, I will make SHEP open source. This means that everyone can work on SHEP, and get SHEP to help them in their lives. I was reviewing the algorithm of SHEP. I was thinking, sometimes a word is a subject, and other times not. SHEP should be able to learn how to tell when words should/shouldn’t be. This is particularly difficult as it is impossible to teach it each time. The annoying thing is, I am currently the only developer of SHEP. Every aspect of SHEP I have to think of and program in. I will continue to improve SHEPs algorithm but until I have a team, SHEP is very limited to my imagination. This is the good thing about open source, it can be shared and bettered, and credit given to me for the original AI (not that I am doing it for credit alone). I may just be depressed because of exams making me have bad thoughts about SHEP, however life is starting to dull down to reality – and reality is working hard doesn’t always make you rich, knowing the right people does. The AI for this version is nearly concluded, I feel like hardware is not	

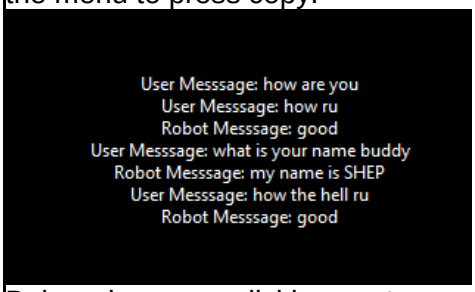
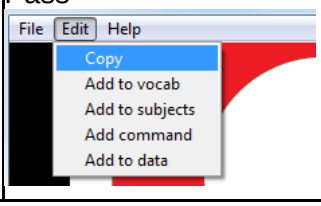
	necessarily needed anymore, SHEP will be software, hardware will be separate. SHEP has a better user interface, better input method, and a better learning algorithm than before.	

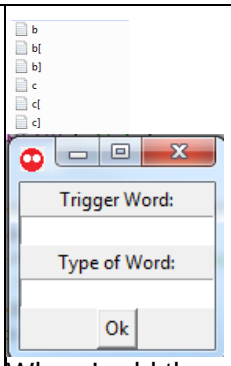
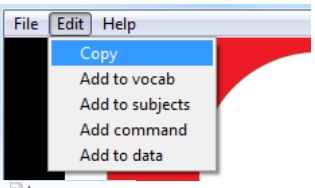
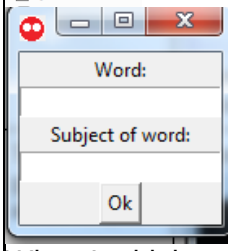
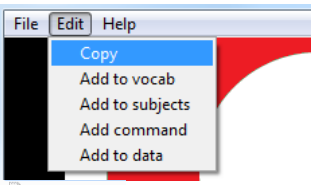
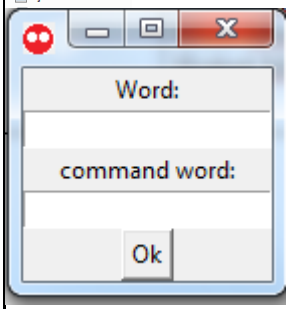
Testing

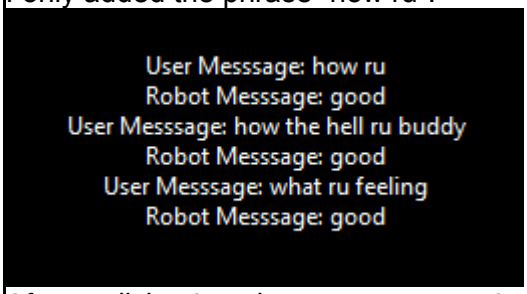
Test results

Test number	Test	Windows PC	Raspberry pi
1	The GUI boots up and works	Pass The system opens as I planned it to. 	This GUI boots up on start-up which means the user does not need to boot it up themselves.
2	The microphone receives input on demand	Pass When I speak the system listens and outputs it on the screen. On top of this it finds an output. 	works
3	If the system does not understand, it gives you a message.	Pass	works

		 This message shows as planned. This test was deemed unnecessary for the later version of this version of the software.	
4	The system outputs your message and the AI's message on screen and stores a log of conversations.	 The system keeps going through the conversation. The system stores a log, however it does not output the log on screen due to space issues.	works
5	The system searches its data for an answer	Pass  I would not get the expected output if this process did not happen	works
6	The system gives out audio output	Pass 	works

		The system opens up an mp3 file to play the string the program told it to play. This test was not needed for the end program, instead it outputs on screen, and to a file.	
7	The system lets you copy the conversation using a function on the menu.	Pass I had a conversation with the AI and went to the menu to press copy.  Below shows me clicking paste: User Message: how are you User Message: how ru Robot Message: good User Message: what is your name buddy Robot Message: my name is SHEP User Message: how the hell ru Robot Message: good	works
8	The menu gives information about the AI and how to use it.	Pass  When I go to the help bar it has a guide option. This pop up comes up with relevant information.	works
9	If the user presses the copy function they can paste the conversation	This test I realised was very similar to test 7. I decided it was unnecessary to carry this out again.	works
10	The user can add trigger words	Pass  	works

		 When I add the words it ends up in the data	
11	The user can add subject words	Pass   When I add the words it ends up in the data	works
12	The user can add command words	Pass   When I add the words it ends up in the data	works

13	The system can have long strings entered in and pick out keywords	Pass I only added the phrase “how ru”:  After realising I spelt message wrong I changed the output. My code was still the same though.  	works
14	The system can communicate with an Arduino to move the system around to make it mobile. Servo learning system will be preferred.	This was specifically developed for the raspberry pi so this function cannot be tested here.	works
15	The AI can learn a series of actions categorized under one action.	No longer needed	works

The raspberry pi has the same effect as the windows on my code, hence everything is the same.

Future ideas

The AI will be put into a core/ head where it can have many sensors and abilities:

The AI should have the ability to see people and know who a person is talking to through eye contact. The AI should have a trigger word for activation and the button; the user will select what they want in the options.

The AI should be able to not just learn through user interaction, but learn through what other people do and how others interact with each other. This gives the AI a more human touch.

The AI will use cameras to recognize objects and search the internet for unknown objects or ask the user to teach them.

The AI should have some sort of safety protocol, for example distances sensors to prevent it from walking into walls or off of edges.

The AI doesn't need a subject of conversation all the time, if you talk about apples and then say a sentence without a subject, the AI will assume you are speaking about apples still.

SHEP can not only be mobile, but connect to a server where a main SHEP AI controls lots of systems. I could experiment with a small model city and lots of cameras reporting to the server to test out if SHEP could be adapted to control a city, search for people in the city, search for number plates, record crime and report it ECT…

If I can make a SHEP which learns from everything around it, this software could be applied in any robotics. SHEP could be exploring other planets as NASA probes and learning about the planet to teach humans all the way to making life easier by controlling cities or being a servant in a household.

SHEP should be able to use GPS or some sort of location system to work out its position in a house or building so that the robot can navigate around the house/ building when completing tasks.

SHEP should be able to use its Ethernet cable to attach to servers and learn on a mass scale. The raspberry Pi has built in networking so it is a possibility. SHEP needs to be applicable to any system and any job just like a human.

Evaluation

I am very happy with the user interface! Usually I cannot draw or design for my life but I am really happy with SHEPs face and expressions. It is a simple design which makes it easy to work with.

I did not add in the action learning ability as I reformed the software to work in a different way.

I did not complete a lot of the tasks I set out to do, however I decided to make SHEP software and leave hardware as an extra for other projects. SHEP will be open source allowing everyone to work on SHEP.

The next version should include:

- a database management system / re – learn information
  - “Jeff and Jas are married” → “Jeff is single”
- the ability to reply based on the topic of conversation
  - “I like grapes” → “do you like them” (subject missing, subject = grapes?)
- the ability to split paragraphs into sentences
  - “how are you it is a very nice day”
    - “how are you”
    - “it is a very nice day
      - “I’m good. Yes it is a nice day!”
- It can set variables of subjects for that day, then when it checks through the variables to see if the subject has been mentioned in conversation. If it hasn’t, it continues through the file search, but if it has it can mention it, or simply mention its covered that topic before. This is all stored on RAM so can be changed.
  - “the weather is nice” variables[weather] = DataHeld[“nice”]