

VOICE AUTOMATED GUI BUILDER

Neel Mani Upadhyay*¹, Yash Pratap Singh*², Archana Singh*³

*^{1,2,3}Students, School Of Computer Science And Engineering, Galgotias University, India.

ABSTRACT

While we take a break for a second and think about how popular voice-managed devices and digital assistants have come to be, it is most effectively important to appear deeper into the VUI design and examine extra. Doing anything manually takes lots of time, but doing it with voice command saves lots of time. but what if we build a VUI for developing a GUI, does it work? Creating a user interface is not just only creating graphics, it's a combination of UX and user behavior. Whether It works or not it's not important, the most important is how to implement this model to create a VUI for creating a GUI. That's part we're looking at in this paper. We're using ML and AI technologies to implement this model and to automate GUI creation we're using PyQt5 in python. But why we're building this model, because VUI is faster than manual, Ease of use, Hands-free, and Eye's-free. The most well-known voice user interface examples are Google Assistant, Amazon's Alexa, etc. Not only AI assistants but also smart devices such as Amazon Echo, Apple Home Pod, and Google Home with VUIs are taking over the market. A recent article from Google called "How voice assistance is reshaping consumer behavior" makes the massive upward push of smart speakers and VUI even greater apparent. According to Google's article, 41% of individuals who personal a voice-controlled speaker experience like speaking to any other person.

I. INTRODUCTION

First, What is GUI? GUI stands for Graphical User Interface, which allows the user to communicate with machines in an easy way. GUI consists of two things, UI and UX. UI is a User Interface, where the elements like button, inputbox, dropdown, image, etc. user sees on the screen comes into the UI. UX is User Experience, suppose there are two buttons called "Save" and "Cancel", "Save" button normally consists of blue color, which signals to the user to be calm and safe and also increases the probability of the user clicking on that button. And the "Cancel" button normally consists of red color, which signals to the user to be alert as it is the color of danger and also decreases the probability of the user clicking on that button. In short, User Experience is measured by how easy or difficult it is to interact with the User Interface. But, interacting with GUI needs attention and it is manual and, we want something that doesn't need to be manual work, It is controlled by our voice, and here VUIs come into the picture. Voice User Interface (VUI) allows users to interact with an app through their voice commands. VUI is faster than manual, Ease of use, Hands-free, and Eye's-free. The world leading five big companies like Google, Facebook, Amazon, Microsoft, Apple are developing voice-activated AI Assistants. The most popular voice-activated AI Assistants are Apple's Siri, Google's Assistant, and Amazon's Alexa and all of these are the best examples of the VUI. Tesla's self-driving car is an advanced implementation of VUI, you only have to tell the place or check points and it will lend you. This all shows that the VUI is the future of technology.

1. HOW DOES A VOICE INTERFACE WORK?

VUIs are implemented by using AI technologies like, Natural Language Processing (NLP), Speech Recognition, Text to Speech Synthesis, Machine Learning, and Deep Learning.

NLP is AI technology which is used to understand humans' natural language by machine. Speech Recognition is the ability to recognise the sentence which is said by humans, and Text to Speech is the ability to speak natural language by machine. Machine learning is a technique to how machines can learn from large amounts of data and predict the next consequences. Deep Learning is a technique that machines can learn by themselves by analyzing the behavior, pattern, color, size, and on the basis of other factors of an object. By using all these combinations we can implement the model to understand user command and take action on it. Now let's discuss what system requirements users have to run this program.

2. HARDWARE & SOFTWARE REQUIREMENT

The software is light weighted and consumes less resources and it's not difficult to meet this requirement because it generally wants basic requirements.

Hardware Requirement -

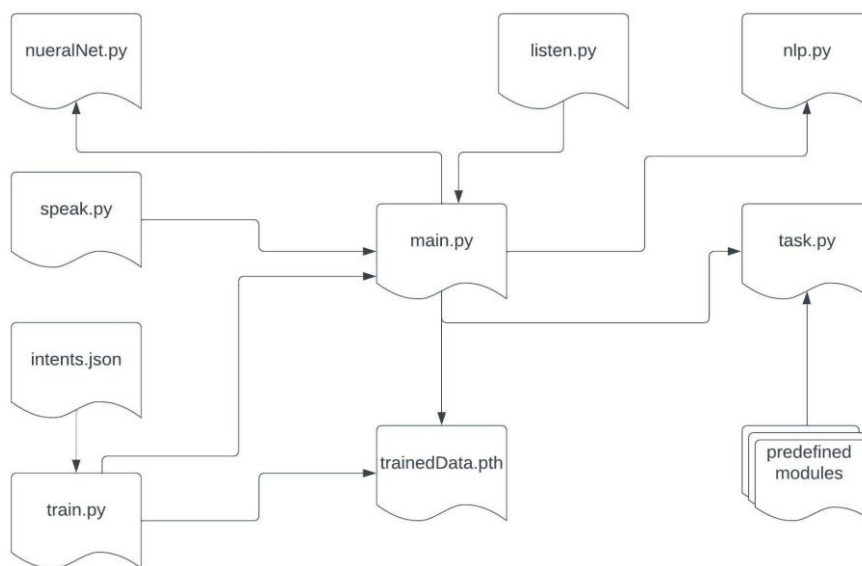
- Intel Dual Core Processor or later
- 4GB RAM or more

Software Requirement -

- Windows 7 or later
- Python 3.5 or later
- Pycharm Community or Any Code Editor
- Python Modules - pyttsx3, PyQt5, pytorch, pywinauto, speech-recognition and nltk

Now, let's discuss the system design of this app and how we can implement this application.

3. SYSTEM DESIGN



This is the DFD diagram for our programs. It has so many things but don't worry, we'll discuss it one by one.

listen.py - This file contains the code of speech recognition (to recognize what the user says).

speak.py - This file contains the code for speaking responses to the user using pyttsx3 module.

nlp.py - This file contains the code for Natural Language Processing (to remove unusable words and focus on the main command).

neuralNet.py - This is a feed forward neural network to input input_size, hidden_size, and num_classes. After that it calculates the rectified linear value of x and returns the value of x.

intents.json - It is a json file which contains tags, patterns, and responses.

- **Tags** defines what kind of task it is.
- **Patterns** defines what kind of pattern should be followed by a user command to perform a specific task.
- **Responses** defines responses on the basis of patterns that forward to the next module.

train.py - This file contains the code for a neural network to train the model from the intents.json file and its created trainedData.pth file when training is completed, because reading speed from the .pth file is so much faster than a normal file.

trainedData.pth - It is a data file which contains the data regarding the intents.json file. If you change or update intents .json then you will delete the old trainedData.pth file and run the train.py again to create a new updated trainedData.pth file.

predefined modules - It is a package which contains the classes and functions regarding automating the Qt Designer for creating UI.

task.py - This file is responsible for actually performing a task, it calls the required function from predefined modules on the basis that the responses come from the neuralNet.py. If the value of x is greater than 0.75 (probability) then it will execute the task.

main.py - This is the main file which is responsible for using all the files and handling the execution of the program files that are discussed above.

First, we have to train the model using the intents.json file by using train.py, and it will create the trainedData.pth file. And make sure the total loss must be less than 0.005 otherwise we have to train the model again. Then we can run the main.py and it calls the listen.py to take command from the user. When user gives their command then it forward that command in nlp.py to remove unusable words and focus on the main words, then it will go to neuralNet.py to process the input on the basis of intents and then it will response to the task.py to actually do the task. task.py identifies the type of response then calls the functions from predefined modules.

Here, we can now understand how the system is actually working but the main thing is that how things happen in predefined modules and what are predefined modules?

4. PREDEFINED MODULES

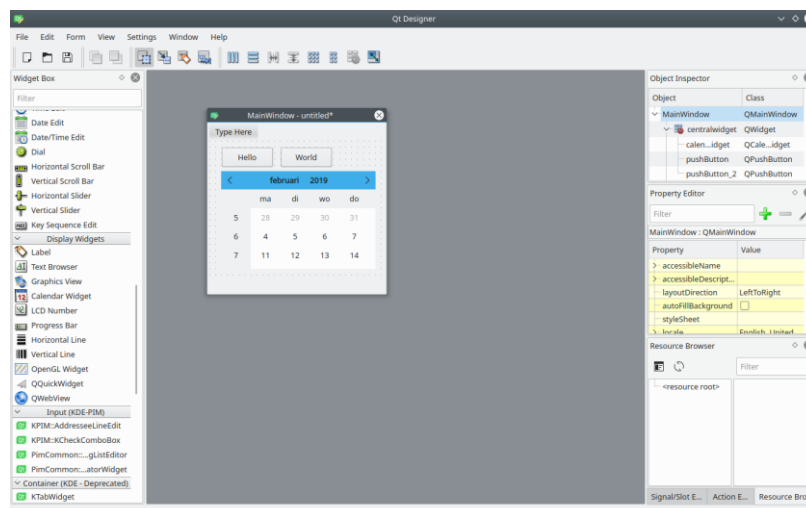
Predefined modules is a python package which contains the classes and functions for controlling and automating the Designer software which is a tool for creating GUI just using drag and drop in PyQt5 library.

PyQt5 - It is a popular python library for creating desktop applications. It is the most powerful framework as it allows users to design modern and animated GUIs from scratch or just using drag and drop.

PyQt5 uses QSS (Qt Style Sheets) for styling Uis. It is similar to CSS (Cascading Style Sheets) but not as powerful as CSS.

pywinauto - It is a python library for sending and controlling the signals of mouse and keyboard in Microsoft Windows GUI. It uses Win32 Api (win32) by default or MS UI Automation (uia) for automating the programs in windows.

Qt Designer - It is an inbuilt tool which helps to create the Uis of the windows application by drag and drop. It is used to build widgets, Main Windows, and Dialogs. It also contains a lot of UI components like PushButton, RadioButton, Check Boxes, LineEdit, Label, Widget, Frame, etc.. Also it contains a Property Editor to customize the UI components as user needs, Object Inspector to track all the components, etc..



By using pywinauto modules, we can create predefined modules which automate the Qt Designer to create the desired UIs by voice commands.

II. IMPLEMENTATION

We've discussed the system requirement, project structure, system design, and required modules to build this project, But the most important question is How? How can users control the application only with their voices? Let's discuss, users can easily give commands to machines to do normal tasks like search something, turn on or off appliances, play songs, etc... All of these are normal one way of giving commands. But, in GUI builder applications it is complex to tell the machine where to place the widget, how to connect elements, and how to maintain the responsiveness of the application when creating UI.

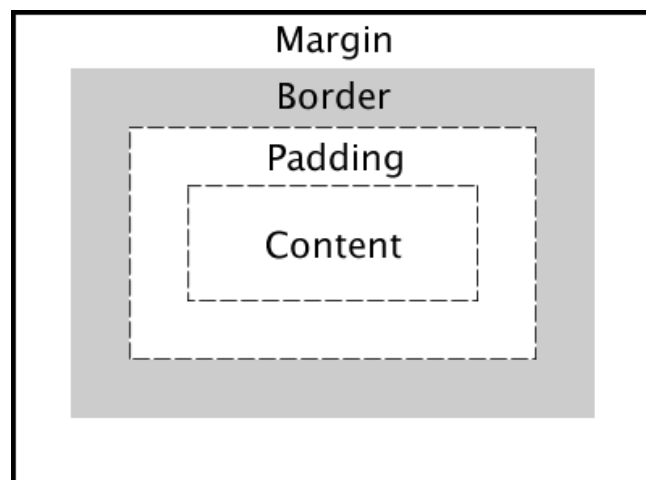
For example - if we create a login/signup form and also we want to make it responsive and we want some label to be at that position which can only be defined by coordinates, but how to tell the machine on what position we can place the label. Is it the coordinate of the UI window or screen coordinates? How to shift the ui component when a specific width of the app is triggered?

And that's the real issue of this research paper. Let's discuss how to solve this problem.

VTAGS

Just like in web development there are many CSS frameworks like Bootstrap, Tailwind Css, etc. which make it easier to style web pages. This framework has predefined classes like p-1, mx-1, etc. used to style the web pages only adding classes names.

We can use the same strategy to fill the gap between user and machine communication in our application. Like intents.json, we have tags, patterns, and responses which are used to decide the task on the basis of user commands. In the same way we can have some predefined tags which helps the user to locate the widget in the correct place. UI components have the same box model as CSS box model.



We can use tags like lp-2 means left padding of 2px, rm-5 means right margin of 5px, etc.. which can help users to define the position correctly.

III. CONCLUSION

VUI based GUI builder is the first VUI based application which can be implemented by using one system design, but there are many more ways to design the VUI system.

In our application, we can discuss the GUI, difference between VUI and GUI, we have discussed hardware and software requirements, and system design. We have discussed the role of each file and directory. And when implementing we have faced the problem of communication gap between user and app, and discussed one solution to fix the communication gap between user and app. But there are a lot more problems which haven't been discussed in this paper but we are continuously working on it.

ACKNOWLEDGEMENT

We have a lot of experience in this project and we have gained a lot of new skills from this project. However, this would not have been possible without the support and help of many people. A heartfelt thank you to all of them. I am very grateful to **Mrs. Archana Singh** for their guidance and continued supervision as well as providing the necessary information regarding the project and am very grateful for their supervision. And also thanks to a friend who offers friendly cooperation and support to help us to complete our project.

IV. REFERENCES

- [1] Diane J Litman and Shimei Pan. 2000. Predicting and adapting to poor speech recognition in a spoken dialogue system. Proceedings of the National Conference on Artificial Intelligence (2000), 722--728.
- [2] Janghee Cho. 2018. Mental Models and Home Virtual Assistants (HVAs). Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18 (2018), 1--6. The Impact of

- User Characteristics and Preferences on Performance with an Unfamiliar Voice User Interface CHI 2019, May 4--9, 2019, Glasgow, Scotland Uk.
- [3] J. Chu-Carroll. 2000. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. Proceedings of the sixth conference on Applied natural language processing 6 (2000), 97--104.
- [4] Aung Pyae and Tapani N. Joelsson. 2018. Investigating the usability and user experiences of voice user interface. Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct - MobileHCI '18 (2018), 127--131.
- [5] Anushay Furqan, Chelsea Myers, and Jichen Zhu. 2017. Learnability through Adaptive Discovery Tools in Voice User Interfaces. Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17 (2017), 1617--1623.
- [6] Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How Do Users Respond to Voice Input Errors?: Lexical and Phonetic Query Reformulation in Voice Search. Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (2013), 143--152.