

ARTIFICIAL INTELLIGENCE
LAB PROJECT SUBMISSION - 2COE28



**Gamofiz- A voice recognition based Rock, Paper and Scissor Game
against Artificial Intelligence.**



Submitted to: Mrs. Chandni Kumari

Submitted by:

Yashika Gupta 102103828

Tanisha Maheshwary 102153037

Kashvi Jain 102153038

Yuvraj Puri 102153041

Project Name:

Our game Gamofiz, understands the voice commands given by the user and plays rock, paper and scissors with them.

Introduction:

This AI game uses speech recognition technology to convert spoken words into text format. The built-in microphone that can pick up the user's voice and send the audio data to the speech recognition algorithm. The code uses various techniques, such as signal processing, and pattern recognition to convert the audio into text. Once the text is extracted, the game identifies the choice (rock, paper or scissors) mentioned in the user's speech. The app then uses automated technology to randomly compete with the user. It is user-friendly and easy to use. The user simply needs to speak their choice, and the code does the rest. This can efficiently be used among people to cure their boredom who do not have anyone else to play with.

Literature Survey:

Rock-paper-scissors is a popular hand game that has been played for generations. With the rise of voice recognition technology, many developers have created games that allow users to play rock-paper-scissors using their voice.

One example of a rock-paper-scissors game that uses voice recognition is "Rock Paper Scissors Game" developed by Mirko Dimartino. This game is available on the App Store for iOS devices and uses Siri's speech recognition technology to detect the user's voice commands. The game features different difficulty levels and allows users to play against the computer or another player.

Another example is "Voice Rock Paper Scissors" developed by BlueHornet. This game is available on the Google Play Store for Android devices and uses Google's Cloud Speech API to detect the user's voice commands. The game features different sound effects and animations to make the gameplay more engaging.

A research paper titled "Voice-Activated Rock-Paper-Scissors Using Deep Learning" by Zhao et al. proposes a system that uses deep learning to detect hand gestures and speech commands to play rock-paper-scissors. The system uses a convolutional neural network (CNN) to recognize hand gestures and a speech recognition algorithm to detect voice commands. The system achieves high accuracy in recognizing hand gestures and voice commands, making the gameplay more intuitive and engaging.

Overall, there are many examples of rock-paper-scissors games that use voice recognition technology, ranging from simple mobile apps to more advanced systems that use deep learning algorithms. These games offer a fun and interactive way to play rock-paper-scissors, and could potentially be used for educational or therapeutic purposes, such as teaching children or helping people with disabilities to improve their communication skills.

Methodology:

1. Step by step working (imported libraries w their info + their commands how to import them, code, working example + project link)

Libraries Imported:

- **pyttsx3**- is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

```
pip install pyttsx3
```

- **PyAudio**- provides Python bindings for PortAudio v19, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms, such as GNU/Linux, Microsoft Windows, and Apple macOS.

```
pip install pyaudio
```

- **SpeechRecognition**- SpeechRecognition incorporates computer science and linguistics to identify spoken words and convert them into text. It allows computers to understand human language.

```
pip install speechrecognition
```

- **Random**- Python has a built-in module that you can use to make random numbers.

```
import random
```

Code:

```
import random

import pyttsx3

import speech_recognition as sr

engine=pyttsx3.init('sapi5')

voices=engine.getProperty('voices')

engine.setProperty('voice',voices[1].id)

def speak(audio):

    engine.say(audio)

    engine.runAndWait()

def takeCommand():

    r=sr.Recognizer()

    with sr.Microphone() as source:

        print("Listening.....")

        r.pause_threshold=1

        audio=r.listen(source)

    try:

        print("Recognising your voice.....")

        query=r.recognize_google(audio,language='en.in')

        print("user said:\n"+ str(query))
```

```
except Exception as e:

    print(e)

    print("Could not recognise your voice\nPlease speak again")

    return query

def game(comp,you):

    if comp==you:

        return None

    elif comp=='rock':

        if you=='scissors':

            return False

        elif you=='paper':

            return True

    elif comp=='paper':

        if you=='rock':

            return False

        elif you=='scissors':

            return True

    elif comp=='scissors':

        if you=='paper':

            return False

        elif you=='rock':

            return True
```

```
def result():

    speak("You chose" +you)

    speak("and computer choosed"+comp)

    print("You chose" +you)

    print("and computer choosed"+comp)


print("*****Welcome to Rock Paper Scissors Game*****")

speak("Welcome to Rock Paper Scissors Game")


while True:

    randno=random.randint(1,9)

    if randno>=1 and randno<=3:

        comp='rock'

    elif randno>=4 and randno<=6:

        comp='paper'

    elif randno>=7 and randno<=9:

        comp='scissors'

    #Taking input from user

    print("Your turn: Speak your choices from")
```

```
    speak("Please speak your choice")

    print("1> Rock")

    print("2> Paper")

    print("3> Scissors")

    print("4> Stop")

    you = takeCommand()

    if 'stop' in you:

        break

    elif you!='rock' and you!='paper' and you!='scissors':

        speak("Invalid choice")

        print("Invalid choice")

    elif 'rock' or 'paper' or 'scissors' in you:

        a=game(comp,you)

        if a==None:

            result()

            print("Draw")

            speak("Draw")

        if a==False:

            result()

            print("lost")

            speak("lost")

        if a==True:
```

```
result()  
  
print("won")  
  
speak("won")
```

Working example:

```
*****Welcome to Rock Paper Scissors Game*****  
Your turn: Speak your choices from  
1> Rock  
2> Paper  
3> Scissors  
4> Stop  
Listening.....  
Recognising your voice.....  
user said:  
paper  
You choosedpaper  
and computer choosedrock  
won
```

```
Listening.....  
Recognising your voice.....  
user said:  
stop  
PS C:\Users\Punit>
```

Project link:

<https://github.com/flytanisha/Voice-Recognition-Based-Game-AI.git>

2. Datasets-

Here are some suggestions for datasets that you can use for training a voice recognition game for Rock Paper Scissors:

Kaggle Rock Paper Scissors Audio Dataset: This is a dataset consisting of 4,000 audio recordings of individuals playing Rock Paper Scissors. The dataset is available in WAV format and was collected by the Kaggle community.

Google Speech Commands Dataset: This is a dataset consisting of over 100,000 audio recordings of common voice commands, including "rock," "paper," and "scissors." The dataset is available in WAV format and was collected by Google.

VoxCeleb2 Dataset: This is a dataset consisting of over 1 million audio recordings of individuals speaking, including celebrities and non-celebrities. While not specifically focused on Rock Paper Scissors, it may be useful for training a general voice recognition model.

UrbanSound8k Dataset: This is a dataset consisting of 8,000 audio recordings of urban sounds, including sounds such as sirens, dogs barking, and car horns. While not directly related to Rock Paper Scissors, it may be useful for training a model to recognize and classify different types of audio.

3. Algorithm -

Here's a simple algorithm for a rock-paper-scissors game that uses voice recognition:

Ask the user to say "rock", "paper", or "scissors".

Use a speech recognition library or API (such as Google Cloud Speech-to-Text) to convert the user's spoken input into text.

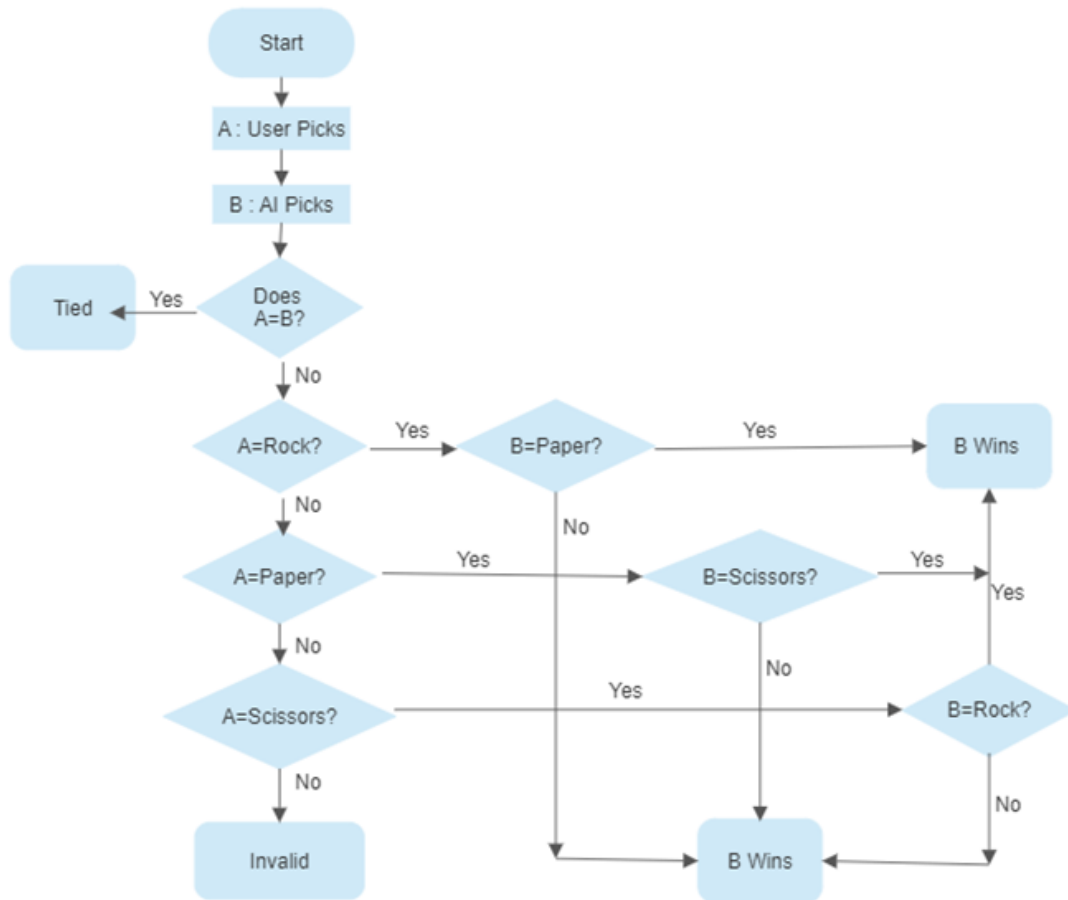
Compare the user's choice to the computer's choice, which should be generated randomly.

Determine the winner based on the standard rock-paper-scissors rules (rock beats scissors, paper beats rock, scissors beat paper).

Display the result and ask the user if they want to play again.

It's important to note that training a voice recognition model for Rock Paper Scissors will likely require a relatively small amount of data compared to other machine learning tasks. Therefore, even a relatively small dataset may be sufficient for this task.

4. Flowchart -



Results and future scope:

The use of voice recognition technology in rock-paper-scissors games has the potential for a wide range of future applications and advancements. Here are some of the potential future scopes of rock-paper-scissors voice recognition games:

Improved accuracy: One of the main challenges in voice recognition technology is achieving high accuracy, especially in noisy environments. Future advancements in speech recognition technology could improve the accuracy of voice recognition in rock-paper-scissors games, making the gameplay more seamless and engaging.

Multiplayer games: Current rock-paper-scissors voice recognition games are limited to single-player or two-player games. Future advancements in voice recognition

technology could enable more players to participate in the game, making it more social and interactive.

Virtual assistants: Voice-activated rock-paper-scissors games could be integrated into virtual assistant platforms such as Alexa or Google Home. Users could play the game using voice commands and compete against other users, making it a fun and engaging way to interact with virtual assistants.

Integration with other technologies: Voice recognition technology could be integrated with other technologies such as augmented reality or virtual reality, creating an immersive gaming experience. Users could use voice commands to control their virtual hands and play rock-paper-scissors in a virtual environment.

Educational applications: Voice-activated rock-paper-scissors games could be used for educational purposes such as teaching children about different hand gestures or improving communication skills for people with disabilities. Games could be designed to be educational and fun, making it an effective tool for learning.

In conclusion, the future scope of rock-paper-scissors voice recognition games is vast and exciting. Advancements in voice recognition technology could make the gameplay more seamless, interactive and engaging, leading to potential applications in various fields.