```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\mithun\anaconda3\lib\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\mithun\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: joblib in c:\users\mithun\anaconda3\lib\site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\mithun\anaconda3\lib\site-packages (from nltk) (2022.7.9)
Requirement already satisfied: tqdm in c:\users\mithun\anaconda3\lib\site-packages (from nltk) (4.65.0)
Requirement already satisfied: colorama in c:\users\mithun\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!pip install wordcloud
```

```
Collecting wordcloud
  Obtaining dependency information for wordcloud from https://files.pythonhosted.org/packages/f5/b0/247159f61c5d5d6647171bef84430b7efad4
  Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl.metadata (3.5 kB)
Requirement already satisfied: numpy>=1.6.1 in c:\users\mithun\anaconda3\lib\site-packages (from wordcloud) (1.24.3)
Requirement already satisfied: pillow in c:\users\mithun\anaconda3\lib\site-packages (from wordcloud) (10.0.1)
Requirement already satisfied: matplotlib in c:\users\mithun\anaconda3\lib\site-packages (from wordcloud) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\mithun\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\mithun\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mithun\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\mithun\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\mithun\anaconda3\lib\site-packages (from matplotlib->wordcloud) (23.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\mithun\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\mithun\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\mithun\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud
Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl (300 kB)
   ---------------------------------------- 0.0/300.2 kB ? eta -:--:--
   ---------------------------------------- 0.0/300.2 kB ? eta -:--:--
   - -------------------------------------- 10.2/300.2 kB ? eta -:--:--
   - -------------------------------------- 10.2/300.2 kB ? eta -:--:--
   --- ------------------------------------ 30.7/300.2 kB 262.6 kB/s eta 0:00:02
   ------- -------------------------------- 61.4/300.2 kB 365.7 kB/s eta 0:00:01
   -------------- ------------------------- 122.9/300.2 kB 554.9 kB/s eta 0:00:01
   ---------------------------------- ----- 256.0/300.2 kB 1.1 MB/s eta 0:00:01
   ---------------------------------------- 300.2/300.2 kB 1.1 MB/s eta 0:00:00
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
```

```
import nltk
from nltk import sent_tokenize
from nltk import word_tokenize
```

```
paragraph = """Games are sometimes played purely for enjoyment, sometimes for achievement or reward as well. They can be played alone, in te
```

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
words=word_tokenize(paragraph)
```

```
print(len(words))
```

```
128
```

```
words
```

```
       'children',
       'playing',
       'a',
       'game',
       'is',
       'deciding',
       'who',
       'is',
       'part',
       'of',
       'their',
       'audience',
       'and',
       'who',
       'is',
       'a',
       'player',
       '.',
       'A',
       'toy',
       'and',
       'a',
       'game',
       'are',
       'not',
       'the',
       'same',
       '.',
       'Toys',
       'generally',
       'allow',
       'for',
       'unrestricted',
       'play',
       'whereas',
       'games',
       'present',
       'rules',
       'for',
       'the',
       'player',
       'to',
       'follow',
       '.']
```

```
#Empty list to store words
words_no_punc= []
#To Remove punctuation marks
for w in words:
    if w.isalpha():
      words_no_punc.append(w.lower())
```

```
words_no_punc
```

```
       'game',
       'are',
       'not',
       'the',
       'same',
       'toys',
       'generally',
       'allow',
       'for',
       'unrestricted',
       'play',
       'whereas',
       'games',
       'present',
       'rules',
       'for',
       'the',
       'player',
       'to',
       'follow']
```

```
print
(len(words_no_punc))
```

```
113
```

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from nltk.corpus import stopwords
#List stopwords
stopwords = set(stopwords.words('english'))
print(stopwords)
```

```
{'most', "weren't", 'below', 'was', "wouldn't", 'herself', 'or', 'while', 'of', 'this', 'between', 'in', 'ma', 'during', 'and', "it's",
```

```
new_words=[]
for w in words_no_punc:
    if w not in stopwords:
        new_words.append(w)
```

```
print(new_words)
```

```
['games', 'sometimes', 'played', 'purely', 'enjoyment', 'sometimes', 'achievement', 'reward', 'well', 'played', 'alone', 'teams', 'onlin
```

```
from nltk.probability import FreqDist
fdist = FreqDist(new_words)
fdist.most_common(10)
```

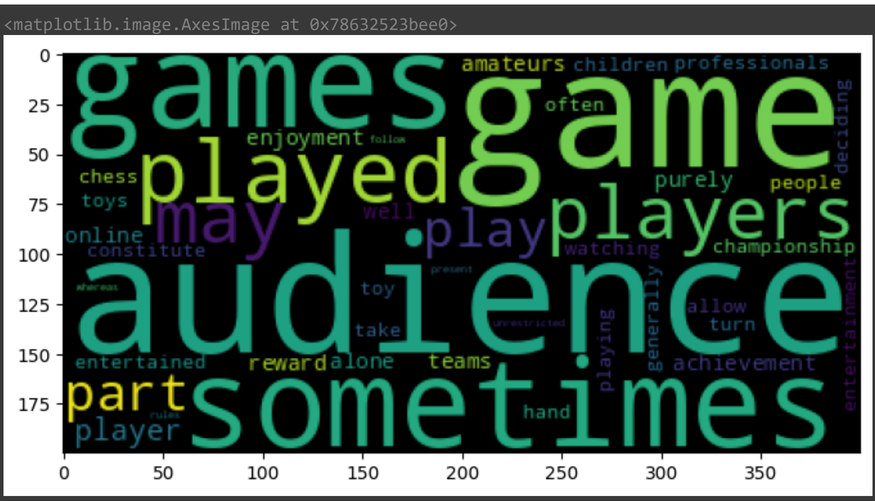
```
[('audience', 3),
 ('game', 3),
 ('games', 2),
 ('sometimes', 2),
 ('played', 2),
 ('players', 2),
 ('may', 2),
 ('play', 2),
 ('part', 2),
 ('player', 2)]
```

```
#Library
from wordcloud import WordCloud
```

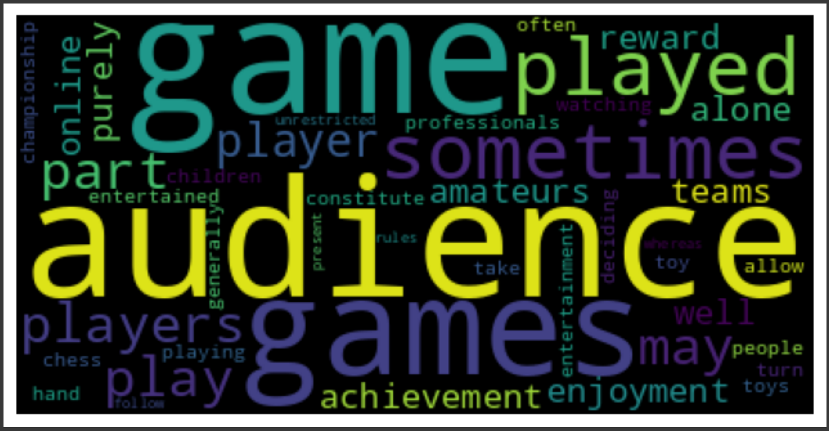
```
#Library to plot the wordcloud
import matplotlib.pyplot as plt
#Generating the worcloud
wordcloud = WordCloud().generate_from_frequencies(fdist)
```

```
#Plot the wordcloud
plt.figure(figsize = (8,8))
plt.imshow(wordcloud)
```

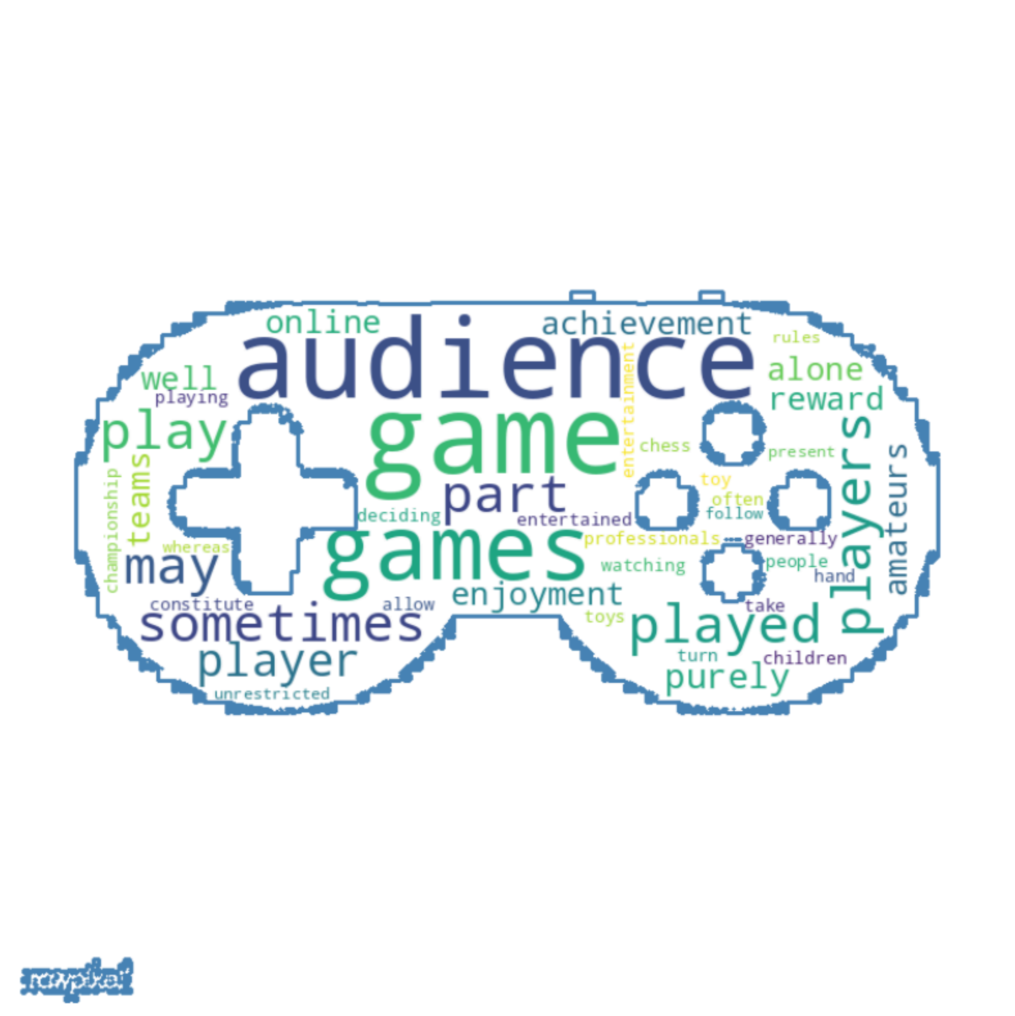
```
<matplotlib.image.AxesImage at 0x78632523bee0>
```



```
#Generating the worcloud
wordcloud = WordCloud().generate_from_frequencies(fdist)
#Plot the wordcloud
plt.figure(figsize = (8,8))
plt.imshow(wordcloud)
#To remove axis value
plt.axis("off")
plt.show()
```



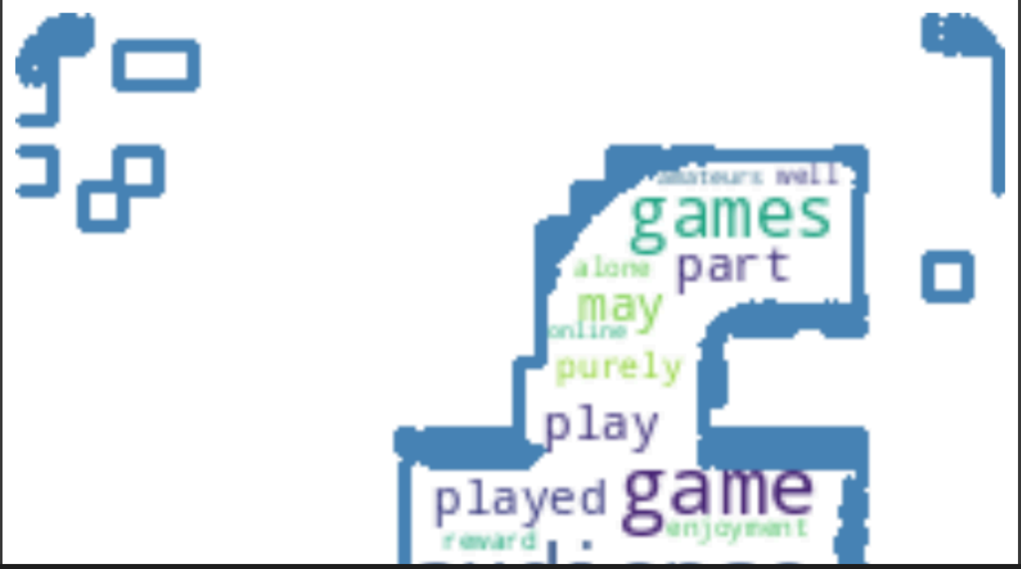
```
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
ship_image = np.array(Image.open("/content/drive/MyDrive/joystick.jpg"))
wordcloud = WordCloud().generate_from_frequencies(fdist)

wordcloud = WordCloud(
    background_color="white",
    mask=ship_image,
    stopwords=stopwords,
    contour_width=3,
    contour_color='steelblue',
    max_words=1000,
    max_font_size=10000,
    random_state=42
).generate_from_frequencies(fdist)

plt.figure(figsize=[10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



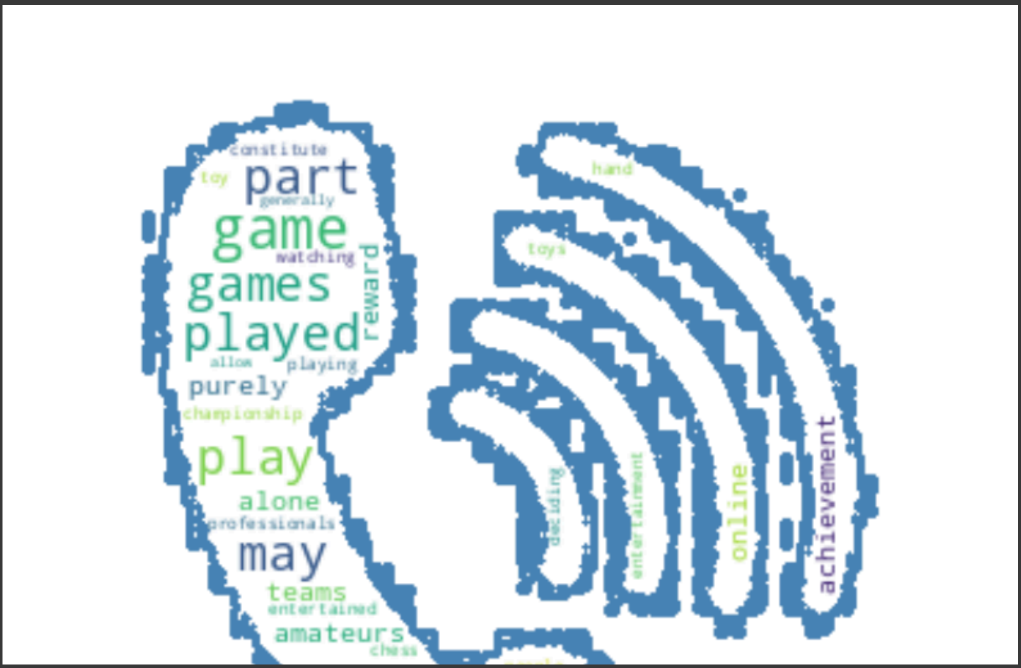
```
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
ship_image = np.array(Image.open("/content/drive/MyDrive/HD-wallpaper-simple-apple-apple-apple-8-apple-9-black-logo-original-phone-red-simpl
wordcloud = WordCloud().generate_from_frequencies(fdist)

wordcloud = WordCloud(
    background_color="white",
    mask=ship_image,
    stopwords=stopwords,
    contour_width=3,
    contour_color='steelblue',
    max_words=1000,
    max_font_size=10000,
    random_state=42
).generate_from_frequencies(fdist)

plt.figure(figsize=[10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

alone
online
enjoyment
play
game
achievement
games
may
amateurs
player
sometimes
part
players
audience
played
well
reward
purely

```
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
ship_image = np.array(Image.open("/content/drive/MyDrive/fb.jpg"))
wordcloud = WordCloud().generate_from_frequencies(fdist)

wordcloud = WordCloud(
    background_color="white",
    mask=ship_image,
    stopwords=stopwords,
    contour_width=3,
    contour_color='steelblue',
    max_words=1000,
    max_font_size=10000,
    random_state=42
).generate_from_frequencies(fdist)

plt.figure(figsize=[10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



```
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
ship_image = np.array(Image.open("/content/drive/MyDrive/phone.jpg"))
wordcloud = WordCloud().generate_from_frequencies(fdist)

wordcloud = WordCloud(
    background_color="white",
    mask=ship_image,
    stopwords=stopwords,
    contour_width=3,
    contour_color='steelblue',
    max_words=1000,
    max_font_size=10000,
    random_state=42
).generate_from_frequencies(fdist)

plt.figure(figsize=[10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```


constitute
toy
part
generally
game
watching
games
reward
played
allow
playing
purely
championship
play
alone
professionals
may
teams
entertained
amateurs
chess
hand
toys
deciding
entertainment
online
achievement