```
import pandas as pd
import re
import spacy
```

```
# Load spaCy English tokenizer
nlp = spacy.load("en_core_web_sm")
# Load the Amazon reviews CSV file
df = pd.read_csv("amazon_reviews.csv")
# Display the first few rows of the "reviewText" column
print(df["reviews.text"].head())
```

```
0    I initially had trouble deciding between the p...
1    Allow me to preface this with a little history...
2    I am enjoying it so far. Great for reading. Ha...
3    I bought one of the first Paperwhites and have...
4    I have to say upfront - I don't like coroporat...
Name: reviews.text, dtype: object
```

```
# Function to clean the text using spaCy
def clean_text_spacy(text):
    if pd.isnull(text): # Handle NaNs
        return []
    text = text.lower() # Convert to lowercase
    text = re.sub(r"[^\w\s]", "", text) # Remove punctuation
    text = text.encode("ascii", "ignore").decode("ascii") # Remove emojis/special characters
    # Tokenize using spaCy
    doc = nlp(text)
    # Remove stopwords and return clean tokens
    tokens = [token.text for token in doc if not token.is_stop and not token.is_punct]
    return tokens
```

```
# Apply the cleaning function to the "reviewText" column
df["cleaned_tokens"] = df["reviews.text"].apply(clean_text_spacy)
# Display the cleaned tokens alongside the original review text
print(df[["reviews.text", "cleaned_tokens"]].head(5))
```

```
                                        reviews.text  \
0  I initially had trouble deciding between the p...
1  Allow me to preface this with a little history...
2  I am enjoying it so far. Great for reading. Ha...
3  I bought one of the first Paperwhites and have...
4  I have to say upfront - I don't like coroporat...

                                      cleaned_tokens
0  [initially, trouble, deciding, paperwhite, voy...
1  [allow, preface, little, history, casual, read...
2  [enjoying, far, great, reading, original, fire...
3  [bought, paperwhites, pleased, constant, compa...
4  [upfront,  , nt, like, coroporate, hermeticall...
```

```
# Flatten the list of all tokens
all_tokens = [token for tokens in df["cleaned_tokens"] for token in tokens]
# Count the frequency of each word using a Counter
from collections import Counter
word_freq = Counter(all_tokens)
# Display the top 15 frequent words in the Amazon reviews
```

```
print("\nTop 15 frequent words in Amazon Reviews:")
print(word_freq.most_common(15))
```

```
Top 15 frequent words in Amazon Reviews:
[('nt', 1951), ('kindle', 1487), ('fire', 1359), ('like', 1235), ('amazon', 1214), ('use', 945), ('great', 798), ('headphones', 780), ('m', 777), ('sound', 774), ('tv', 748), (' '
```