```
import numpy as np

DataFrame = None
Columns = []
def PandasOneHotEncodeNumpy(DataFrame, Columns):
    # Initialize output matrix as None (will be created when we process first column)
    OutNumpyMat = None
    # Initialize list to store column names for the encoded features
    columnNames = []

    # Loop through each column that we want to one-hot encode
    for col in Columns:
        unique_values = sorted(DataFrame[col].unique())
        one_hot = (DataFrame[col].values[:, None] == unique_values).astype(int)
        one_hot = one_hot[:, :-1]

        if OutNumpyMat is None:
            OutNumpyMat = one_hot
        else:
            OutNumpyMat = np.hstack((OutNumpyMat, one_hot))

        columnNames.extend([f"{col}_{val}" for val in unique_values[:-1]])

    return OutNumpyMat, columnNames
```

```
import pandas as pd

# Create a test DataFrame with categorical data
test_data = {
    'Color': ['Red', 'Blue', 'Green', 'Red', 'Blue', 'Green', 'Red'],
    'Size': ['Small', 'Medium', 'Large', 'Small', 'Large', 'Medium', 'Small'],
    'Category': ['A', 'B', 'A', 'C', 'B', 'A', 'C'],
    'Price': [10.5, 25.0, 15.5, 12.0, 30.0, 18.0, 11.5]  # Numerical column for reference
}

DataFrame = pd.DataFrame(test_data)
print("Test DataFrame:")
print(DataFrame)
print("\nDataFrame shape:", DataFrame.shape)
print("Data types:")
print(DataFrame.dtypes)
```

```
Test DataFrame:
   Color    Size Category  Price
0    Red   Small        A   10.5
1   Blue  Medium        B   25.0
2  Green   Large        A   15.5
3    Red   Small        C   12.0
4   Blue   Large        B   30.0
5  Green  Medium        A   18.0
6    Red   Small        C   11.5

DataFrame shape: (7, 4)
Data types:
Color        object
Size         object
Category     object
Price       float64
dtype: object
```

```
# Test the one-hot encoding function
Columns = ['Color', 'Size', 'Category']  # Categorical columns to encode

# Call your function
encoded_matrix, column_names = PandasOneHotEncodeNumpy(DataFrame, Columns)

print("Original DataFrame:")
print(DataFrame)
print("\nOne-hot encoded matrix shape:", encoded_matrix.shape)
print("One-hot encoded matrix:")
print(encoded_matrix)
print("\nColumn names after encoding:")
print(column_names)

# Convert to Float32 as requested
```

```
encoded_matrix_float32 = encoded_matrix.astype(np.float32)
print("\nData type of encoded matrix:", encoded_matrix_float32.dtype)
```

```
Original DataFrame:
   Color    Size Category  Price
0    Red   Small        A   10.5
1   Blue  Medium        B   25.0
2  Green   Large        A   15.5
3    Red   Small        C   12.0
4   Blue   Large        B   30.0
5  Green  Medium        A   18.0
6    Red   Small        C   11.5

One-hot encoded matrix shape: (7, 6)
One-hot encoded matrix:
[[0 0 0 0 1 0]
 [1 0 0 1 0 1]
 [0 1 1 0 1 0]
 [0 0 0 0 0 0]
 [1 0 1 0 0 1]
 [0 1 0 1 1 0]
 [0 0 0 0 0 0]]

Column names after encoding:
['Color_Blue', 'Color_Green', 'Size_Large', 'Size_Medium', 'Category_A', 'Category_B']

Data type of encoded matrix: float32
```