In [59]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('vgsales.csv')
df.head()
```

Out[59]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sa
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10

Data info

In [48]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Rank          16598 non-null  int64
 1   Name          16598 non-null  object
 2   Platform      16598 non-null  object
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object
 5   Publisher     16540 non-null  object
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

In [50]:
```
df.describe()
```

Out[50]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_S
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570

In [51]:
```
df.shape
```

Out[51]:
```
(16598, 11)
```

In [52]:
```
df.columns
```

Out[52]:
```
Index(['Rank', 'Name', 'Platform', 'Year', 'Genre', 'Publisher', 'NA_Sales',
       'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales'],
      dtype='object')
```

Missing values

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
Rank              0
Name              0
Platform          0
Year            271
Genre             0
Publisher        58
NA_Sales          0
EU_Sales          0
JP_Sales          0
Other_Sales       0
Global_Sales      0
dtype: int64
```

Checking how many columns null will delete

In [18]:
```
df.dropna().shape
```

Out[18]:
```
(16291, 11)
```

In [19]:
```
df[df['Year'].isnull()].head()
df[df['Publisher'].isnull()].head()
```

Out[19]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_
470	471	wwe Smackdown vs. Raw 2006	PS2	NaN	Fighting	NaN	1.57	1.02	
1303	1305	Triple Play 99	PS	NaN	Sports	NaN	0.81	0.55	
1662	1664	Shrek / Shrek 2 2-in-1 Gameboy Advance Video	GBA	2007.0	Misc	NaN	0.87	0.32	
2222	2224	Bentley's Hackpack	GBA	2005.0	Misc	NaN	0.67	0.25	
3159	3161	Nicktoons Collection: Game Boy Advance Video V...	GBA	2004.0	Misc	NaN	0.46	0.17	

Drop rows with missing values

In [21]:
```
df_cleaned = df.dropna()
```

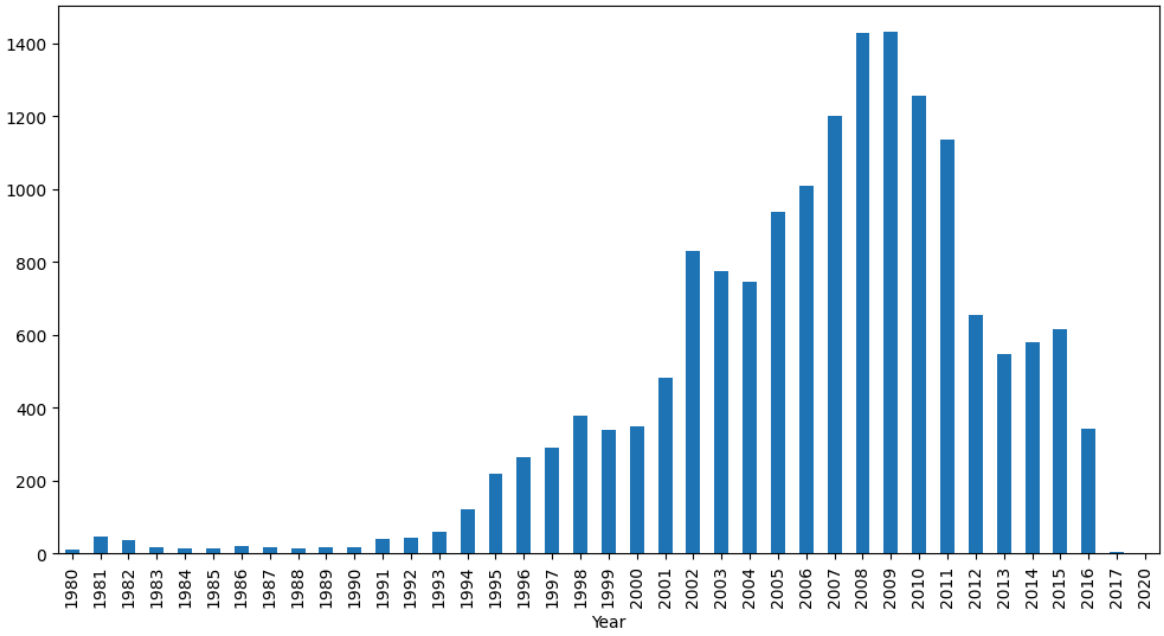
In [22]:
```
df_cleaned.reset_index(drop=True, inplace=True)
```

Converting the date from decimal to integer

In [54]:
```
df_cleaned = df.dropna().copy()
df_cleaned['Year'] = df_cleaned['Year'].astype(int)
```

In [56]:
```
df_cleaned['Year'].value_counts().sort_index().plot(kind='bar', figsize=(12, 6))
```

Out[56]:
```
<Axes: xlabel='Year'>
```



Duplicate vlaues

In [23]:
```
df.duplicated().sum()
```

Out[23]:
```
np.int64(0)
```

Top selling games globally

In [26]:
```
df.sort_values('Global_Sales', ascending=False).head(10)
```

Out[26]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sa
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10
5	6	Tetris	GB	1989.0	Puzzle	Nintendo	23.20	2.26	4
6	7	New Super Mario Bros.	DS	2006.0	Platform	Nintendo	11.38	9.23	6
7	8	Wii Play	Wii	2006.0	Misc	Nintendo	14.03	9.20	2
8	9	New Super Mario Bros. Wii	Wii	2009.0	Platform	Nintendo	14.59	7.06	4
9	10	Duck Hunt	NES	1984.0	Shooter	Nintendo	26.93	0.63	6

Which platform released most games

In [27]:
```
df['Platform'].value_counts().head(10)
```

Out[27]:
```
Platform
DS      2163
PS2     2161
PS3     1329
Wii     1325
X360    1265
PSP     1213
PS      1196
PC       960
XB       824
GBA      822
Name: count, dtype: int64
```

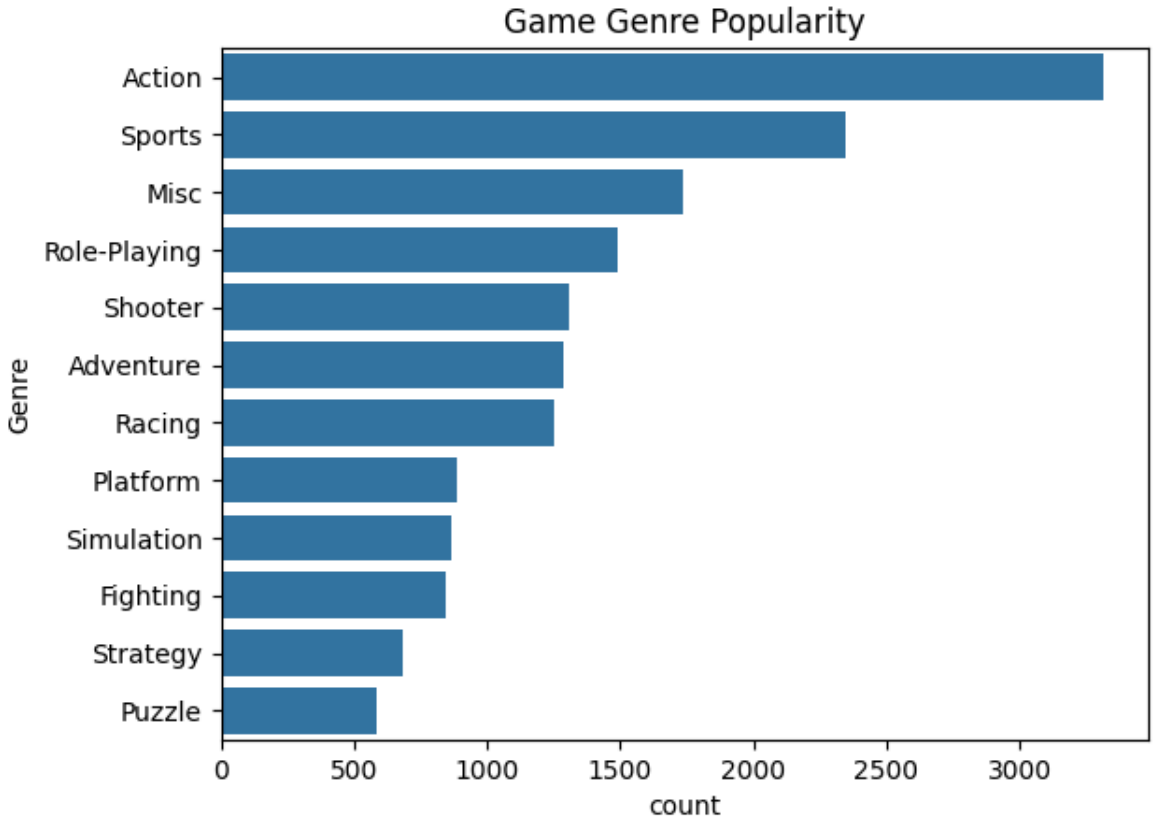
In [28]:
```
df['Publisher'].value_counts().head(10)
```

Out[28]:
```
Publisher
Electronic Arts                 1351
Activision                       975
Namco Bandai Games               932
Ubisoft                          921
Konami Digital Entertainment     832
THQ                              715
Nintendo                         703
Sony Computer Entertainment      683
Sega                             639
Take-Two Interactive             413
Name: count, dtype: int64
```

Genre popularity

In [44]:
```
sns.countplot(y='Genre', data=df, order=df['Genre'].value_counts().index)
plt.title('Game Genre Popularity')
```

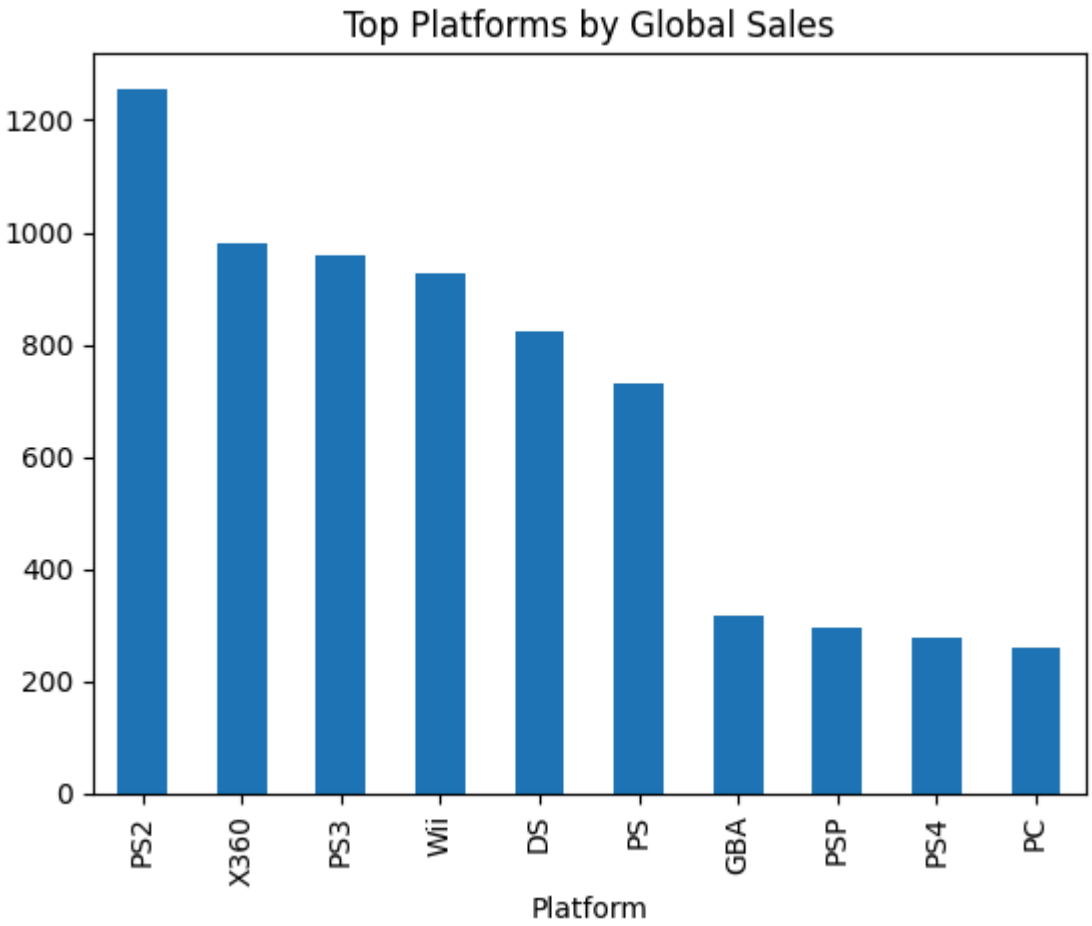
Out[44]:
```
Text(0.5, 1.0, 'Game Genre Popularity')
```



global salesby platform

In [46]:
```
platform_sales = df.groupby('Platform')['Global_Sales'].sum().sort_values(ascend
platform_sales.plot(kind='bar')
plt.title('Top Platforms by Global Sales')
```

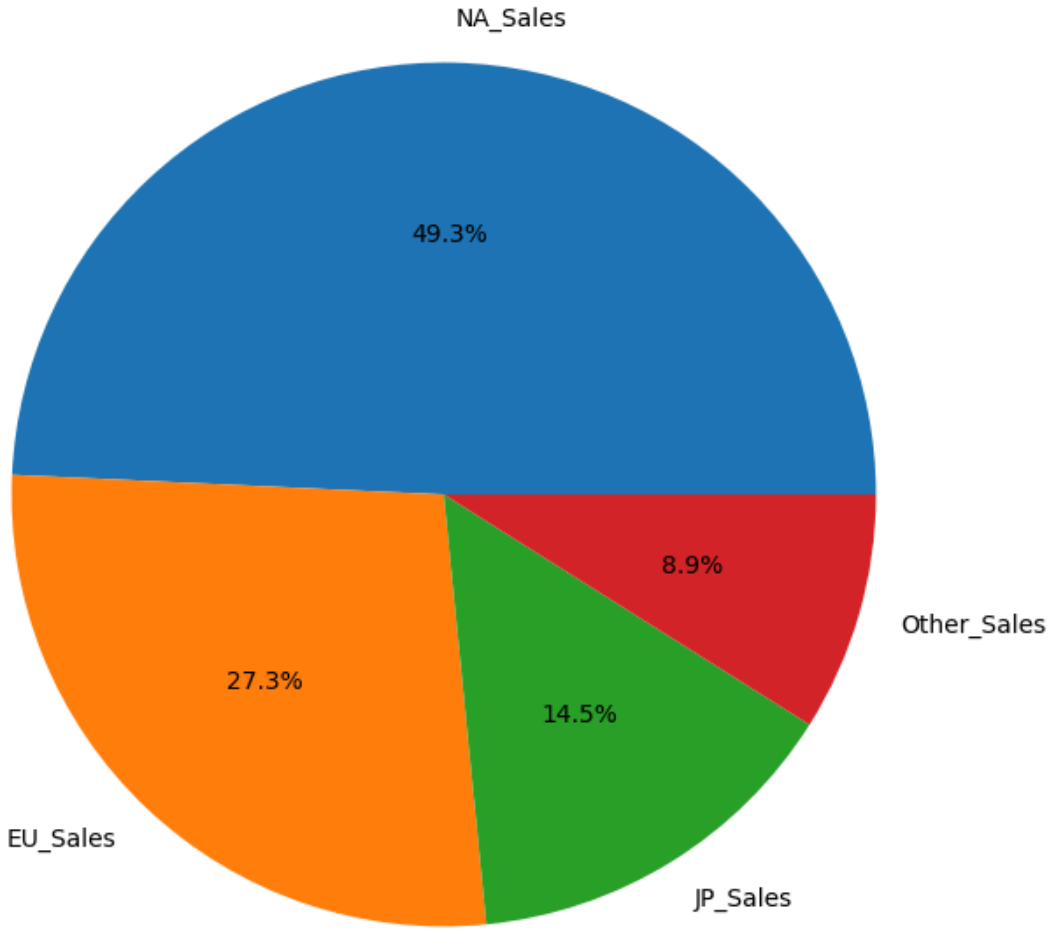
Out[46]:
```
Text(0.5, 1.0, 'Top Platforms by Global Sales')
```



Region-wise sales comparison

In [57]:
```
region_sales = df[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']].sum()
region_sales.plot(kind='pie', autopct='%1.1f%%', figsize=(8, 8))
```

Out[57]: <Axes: >



In [58]:
```
df.to_csv('cleaned_vgsales.csv', index=False)
```

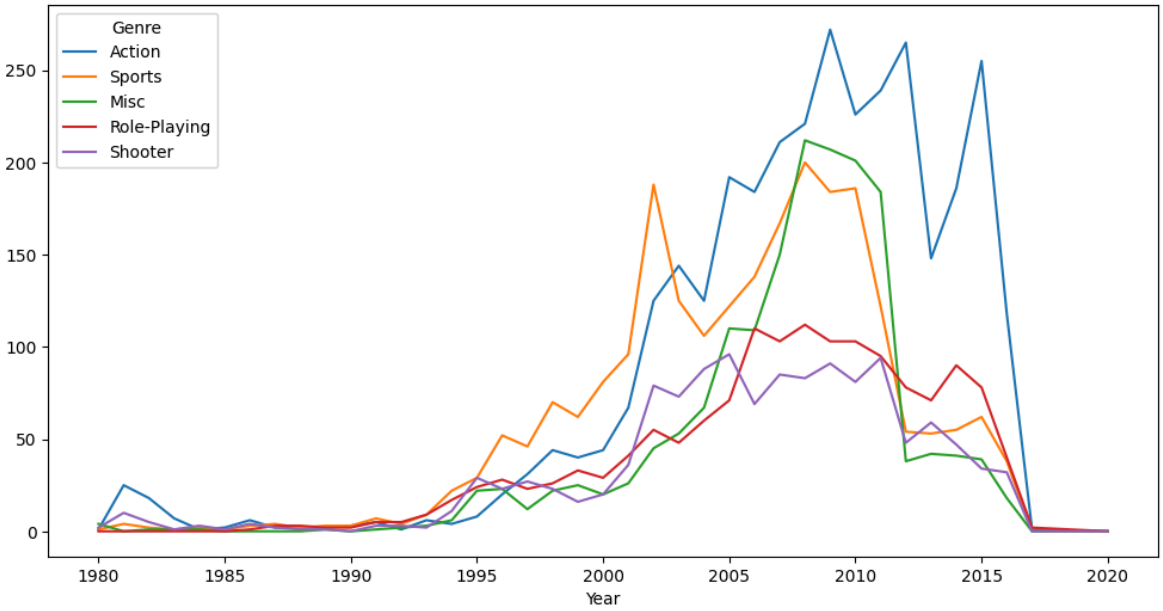
Count how many games of each genre were released each year

In [61]:
```
genre_trend = df_cleaned.groupby(['Year', 'Genre']).size().unstack().fillna(0)
```

Plot only the top 5 genres by total releases

In [63]:
```
top_genres = genre_trend.sum().sort_values(ascending=False).head(5).index
genre_trend[top_genres].plot(figsize=(12,6))
```

Out[63]:
```
<Axes: xlabel='Year'>
```



Sum global sales by year and publisher

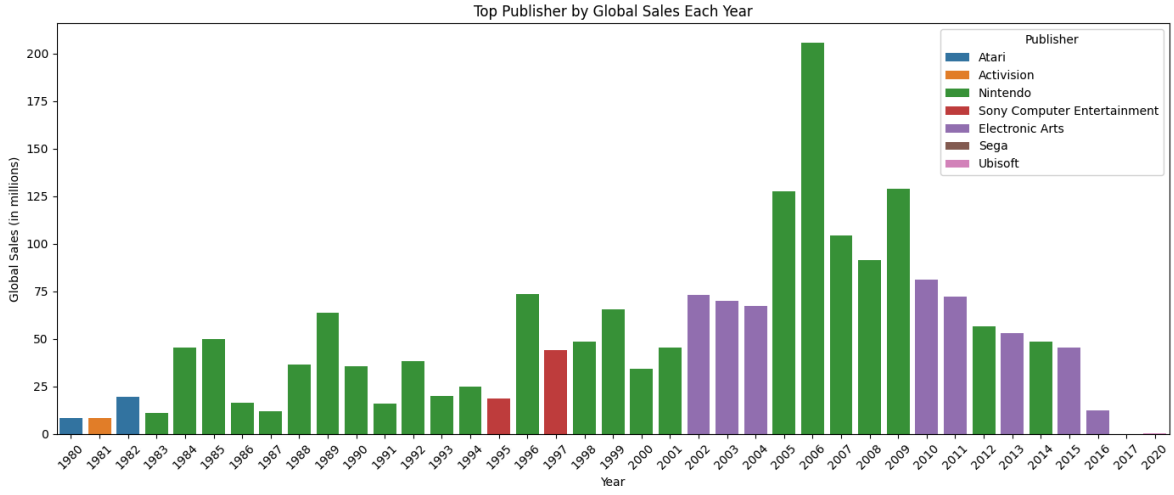
In [73]:
```
publisher_sales_by_year = df_cleaned.groupby(['Year', 'Publisher'])['Global_Sale
```

Get the top publisher for each year

In [74]:
```
top_publishers = publisher_sales_by_year.loc[publisher_sales_by_year.groupby('Ye
```

In [72]:
```
plt.figure(figsize=(14,6))
sns.barplot(data=top_publishers.sort_values('Year'), x='Year', y='Global_Sales',

plt.title('Top Publisher by Global Sales Each Year')
plt.xticks(rotation=45)
plt.ylabel('Global Sales (in millions)')
plt.tight_layout()
plt.show()
```



In [ ]: