In [1]:

```
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import tqdm
from sklearn.linear_model import LinearRegression
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
print("选择Video Game Sales数据集进行电子游戏销售分析")
```

选择Video Game Sales数据集进行电子游戏销售分析

In [3]:

```
data = pd.read_csv('../round3dataset/vgsales.csv')
print('属性类别数:', len(data.columns))
print('总行数:', len(data))
print('示例数据:')
data.head(5)
```

属性类别数: 11
总行数: 16598
示例数据:

Out[3]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	C
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	

In [4]:

```
print('提取每一列属性以及名称')
num_fields = data.select_dtypes(include=np.number).columns.values
nom_fields = data.select_dtypes(exclude=np.number).columns.values
all_fields=[]
for field in nom_fields:
    all_fields.append(field)
for field in num_fields:
    all_fields.append(field)
print("所有属性:", all_fields)
print('标称属性:', nom_fields)
print('数值属性:', num_fields)
print(data.shape," ",nom_fields.shape," ",num_fields.shape)
```

```
提取每一列属性以及名称
所有属性: ['Name', 'Platform', 'Genre', 'Publisher', 'Rank', 'Year', 'NA_Sale
s', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales']
标称属性: ['Name' 'Platform' 'Genre' 'Publisher']
数值属性: ['Rank' 'Year' 'NA_Sales' 'EU_Sales' 'JP_Sales' 'Other_Sales'
 'Global_Sales']
(16598, 11)   (4,)   (7,)
```

In [5]:

```
print("查看每个属性的缺失值统计：")
for field in all_fields:
    print(field,":",data[field].isnull().sum())
```

```
查看每个属性的缺失值统计：
Name : 0
Platform : 0
Genre : 0
Publisher : 58
Rank : 0
Year : 271
NA_Sales : 0
EU_Sales : 0
JP_Sales : 0
Other_Sales : 0
Global_Sales : 0
```

In [6]:

```
print('原始数据行数:', len(data))
data = data.dropna(how='any')
print('将缺失部分剔除后数据行数:', len(data))
```

```
原始数据行数: 16598
将缺失部分剔除后数据行数: 16291
```

In [7]:

```
print("查看最受欢迎游戏的排行")
data_rank = data[['Rank', 'Name']].sort_values(by='Rank')
data_rank.head()
```

查看最受欢迎游戏的排行

Out[7]:

	Rank	Name
0	1	Wii Sports
1	2	Super Mario Bros.
2	3	Mario Kart Wii
3	4	Wii Sports Resort
4	5	Pokemon Red/Pokemon Blue

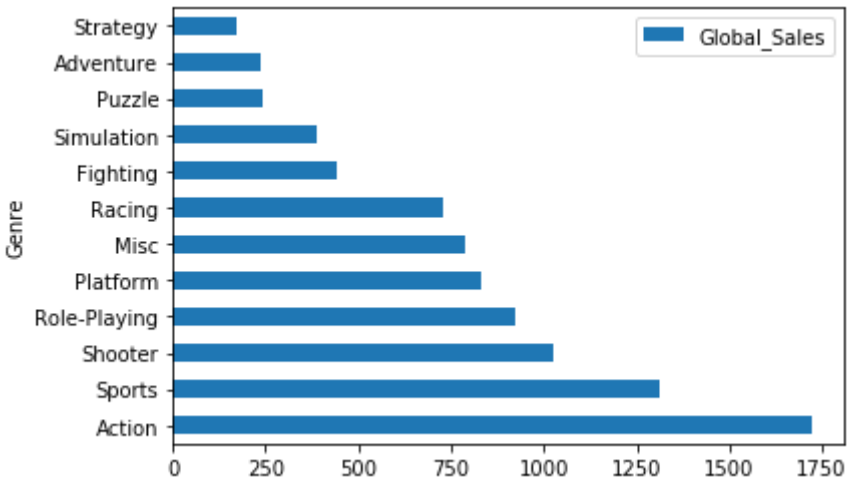
In [8]:

```
print("通过统计相同游戏类型的全球销售额来进行分析。")
data_genre=data[['Genre', 'Global_Sales']].groupby('Genre').sum().sort_values(by='Global_Sa
data_genre.plot.barh()
```

通过统计相同游戏类型的全球销售额来进行分析。

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20a11433978>
```



In [9]:

```
print("可见Sports与Action类型的游戏在全球售额里处于最受欢迎的类型。")
```

可见Sports与Action类型的游戏在全球售额里处于最受欢迎的类型。

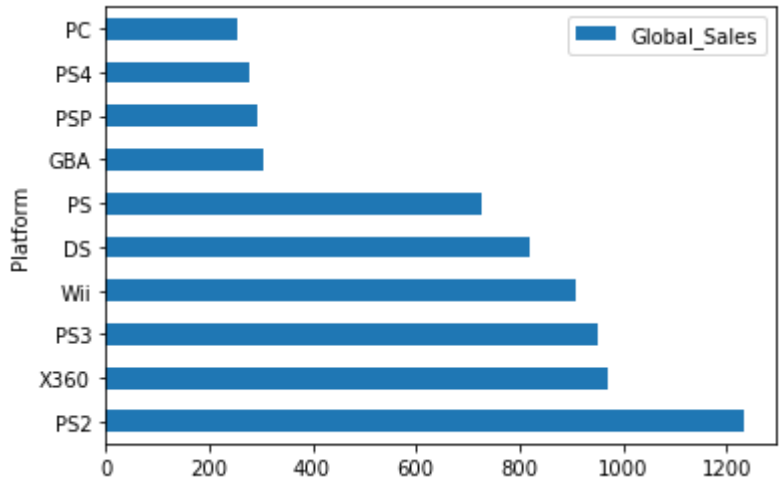
In [10]:

```
data_platform=data[['Platform', 'Global_Sales']].groupby('Platform').sum().sort_values(by='
print('游戏发布平台总数：', len(data_platform))
data_platform.head(10).plot.barh()
```

游戏发布平台总数： 31

Out[10]:

<matplotlib.axes._subplots.AxesSubplot at 0x20a11bc58d0>



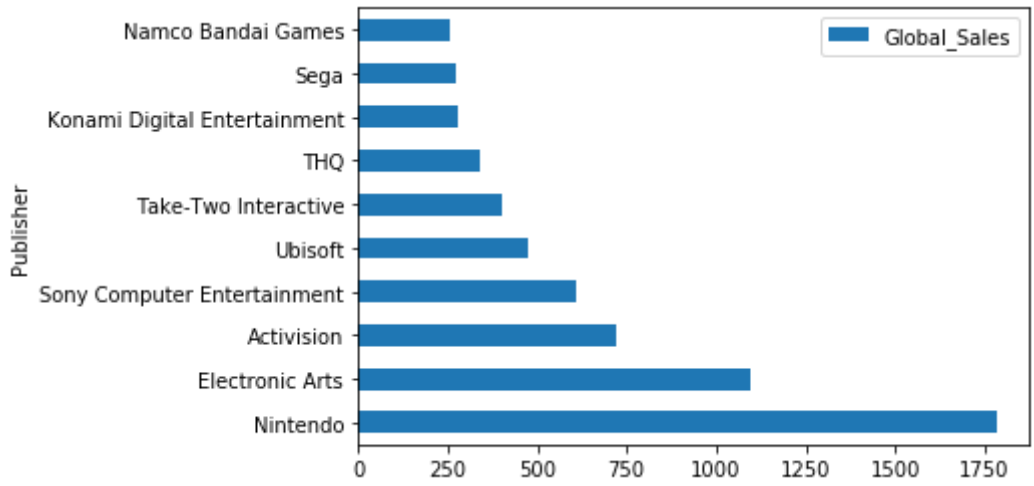
In [11]:

```
data_publisher=data[['Publisher', 'Global_Sales']].groupby('Publisher').sum().sort_values(b
print('游戏发行人总数：', len(data_publisher))
data_publisher.head(10).plot.barh()
```

游戏发行人总数： 576

Out[11]:

<matplotlib.axes._subplots.AxesSubplot at 0x20a11c694e0>



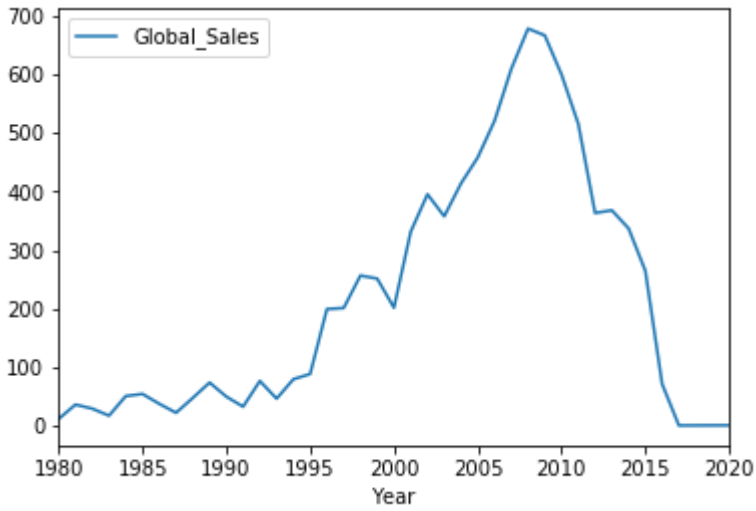
In [12]:

```
print("预测每年电子游戏销售额")
print("统计每年电子游戏销售额信息")
data_year=data[['Year', 'Global_Sales']].groupby('Year').sum().sort_values(by='Year')
data_year.plot()
```

预测每年电子游戏销售额
统计每年电子游戏销售额信息

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20a11cf5748>
```



In [13]:

```
print("建立线性回归模型来预测每年电子游戏销售额")
x = data['Year']
y = data['Global_Sales']
x = np.array(x).reshape(-1, 1)
model=LinearRegression()
model.fit(x, y)
```

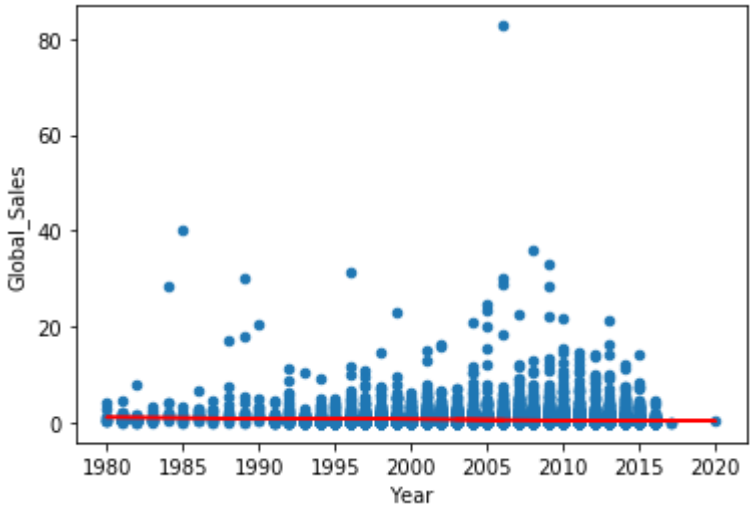
建立线性回归模型来预测每年电子游戏销售额

Out[13]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [14]:

```
data.plot(kind="scatter", x='Year', y='Global_Sales')
plt.plot(x, model.predict(x), 'r-')
plt.show()
```



In [15]:

```
print("可见预测模型出现了一条直线，但是从之前的折线图来看数据刚开始呈上升趋势后来才下降,因此数据存在
print("从电子游戏销售额的折线图中可见在2020年周围销售额为0,这可能是导致错误的原因")
print("查看电子游戏销售额每年销售的具体值")
data_year
```

```
可见预测模型出现了一条直线，但是从之前的折线图来看数据刚开始呈上升趋势后来才下降,因
此数据存在一些问题
从电子游戏销售额的折线图中可见在2020年周围销售额为0,这可能是导致错误的原因
查看电子游戏销售额每年销售的具体值
```

Out[15]:

Year	Global_Sales
1980.0	11.38
1981.0	35.77
1982.0	28.86
1983.0	16.79
1984.0	50.36
1985.0	53.94
1986.0	37.07
1987.0	21.74
1988.0	47.22
1989.0	73.45
1990.0	49.39
1991.0	32.23
1992.0	76.16
1993.0	45.98
1994.0	79.17
1995.0	88.11
1996.0	199.15
1997.0	200.98
1998.0	256.47
1999.0	251.27
2000.0	201.56
2001.0	331.47
2002.0	395.52
2003.0	357.85
2004.0	414.01
2005.0	458.51
2006.0	521.04
2007.0	609.92

Year	Global_Sales
2008.0	678.90
2009.0	667.30
2010.0	600.29
2011.0	515.80
2012.0	363.49
2013.0	368.11
2014.0	337.03
2015.0	264.44
2016.0	70.90
2017.0	0.05
2020.0	0.29

In [16]:

```
print("可见在2016年之后，销售额逐渐变为0，显然这是污点数据。将其剔除掉...")
data_year=data_year.drop([2017, 2020]).reset_index()
```

可见在2016年之后，销售额逐渐变为0，显然这是污点数据。将其剔除掉...

In [17]:

```
print("再次建立线性预测模型")
x = data_year['Year']
y = data_year['Global_Sales']
x = np.array(x).reshape(-1, 1)
model=LinearRegression()
model.fit(x, y)
```

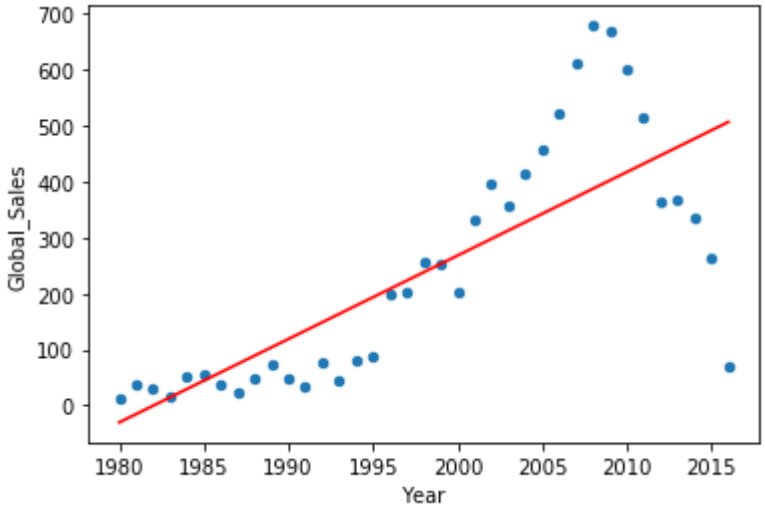
再次建立线性预测模型

Out[17]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [18]:

```
data_year.plot(kind="scatter", x='Year', y='Global_Sales')
plt.plot(x, model.predict(x), 'r-')
plt.show()
```



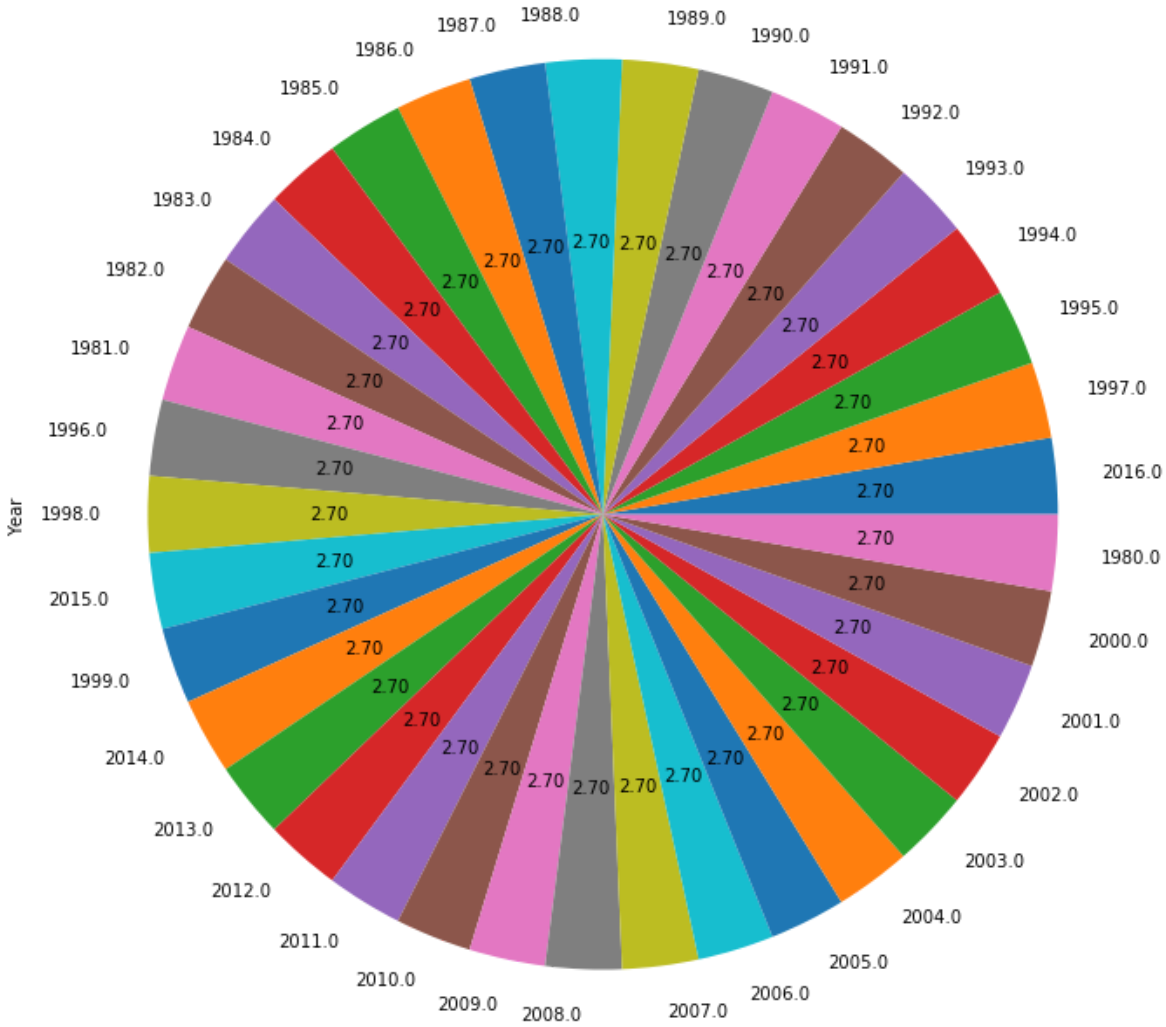
In [19]:

```
print("画出不同年限的电子游戏销售额数量饼图")
label=[]
for key in data_year['Year'].value_counts().index:
    label.append(key)
data_year['Year'].value_counts().plot.pie(labels=label,
                 autopct='%.2f', fontsize=10,figsize=(12, 12))
```

画出不同年限的电子游戏销售额数量饼图

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20a11d9c898>
```



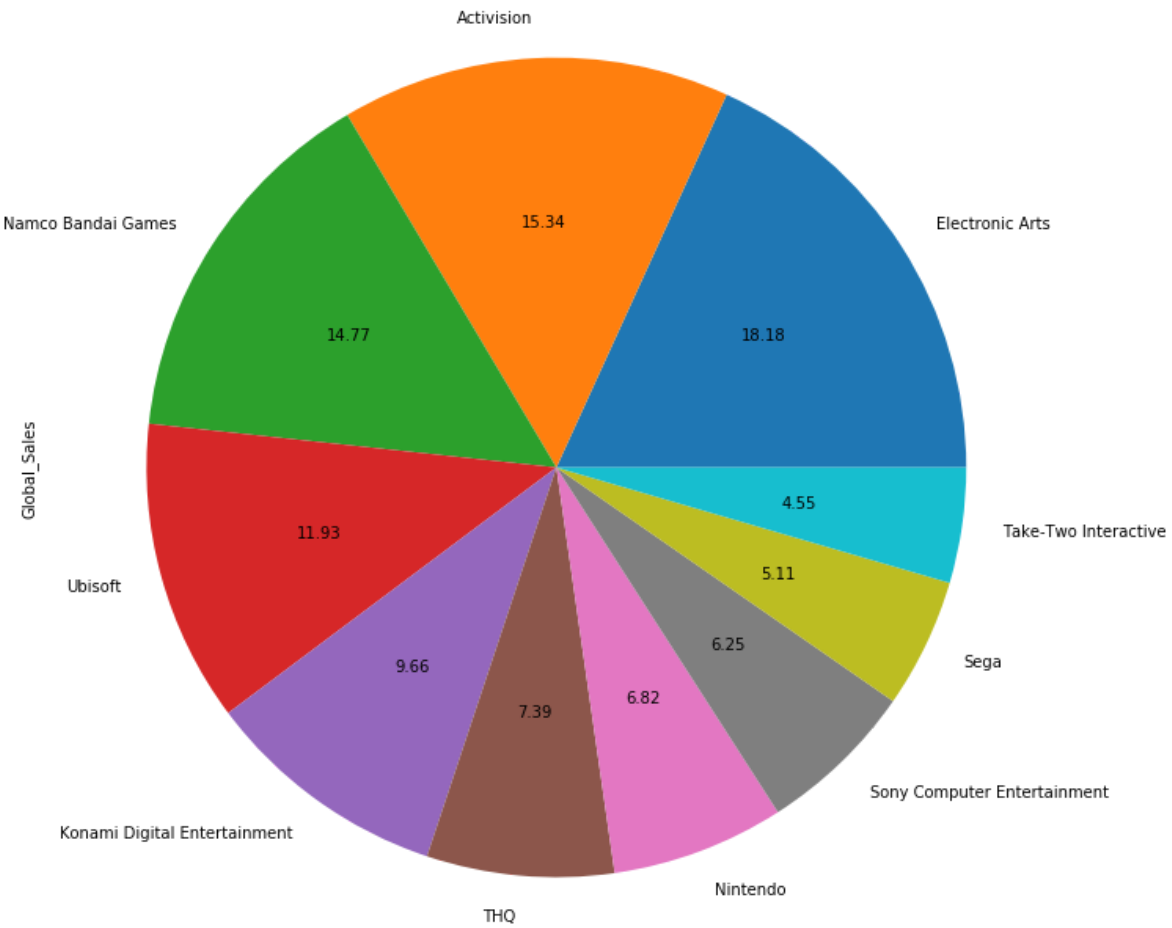
In [27]:

```
print("画出不同出版社的电子游戏销售额前10数量饼图")
label=[]
for key in data['Publisher'].value_counts().index:
    label.append(key)
data_publisher['Global_Sales'].value_counts().head(10).plot.pie(labels=label,
                autopct='%.2f', fontsize=10,figsize=(12, 12))
```

画出不同出版社的电子游戏销售额数量饼图

Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20a11f26f28>
```



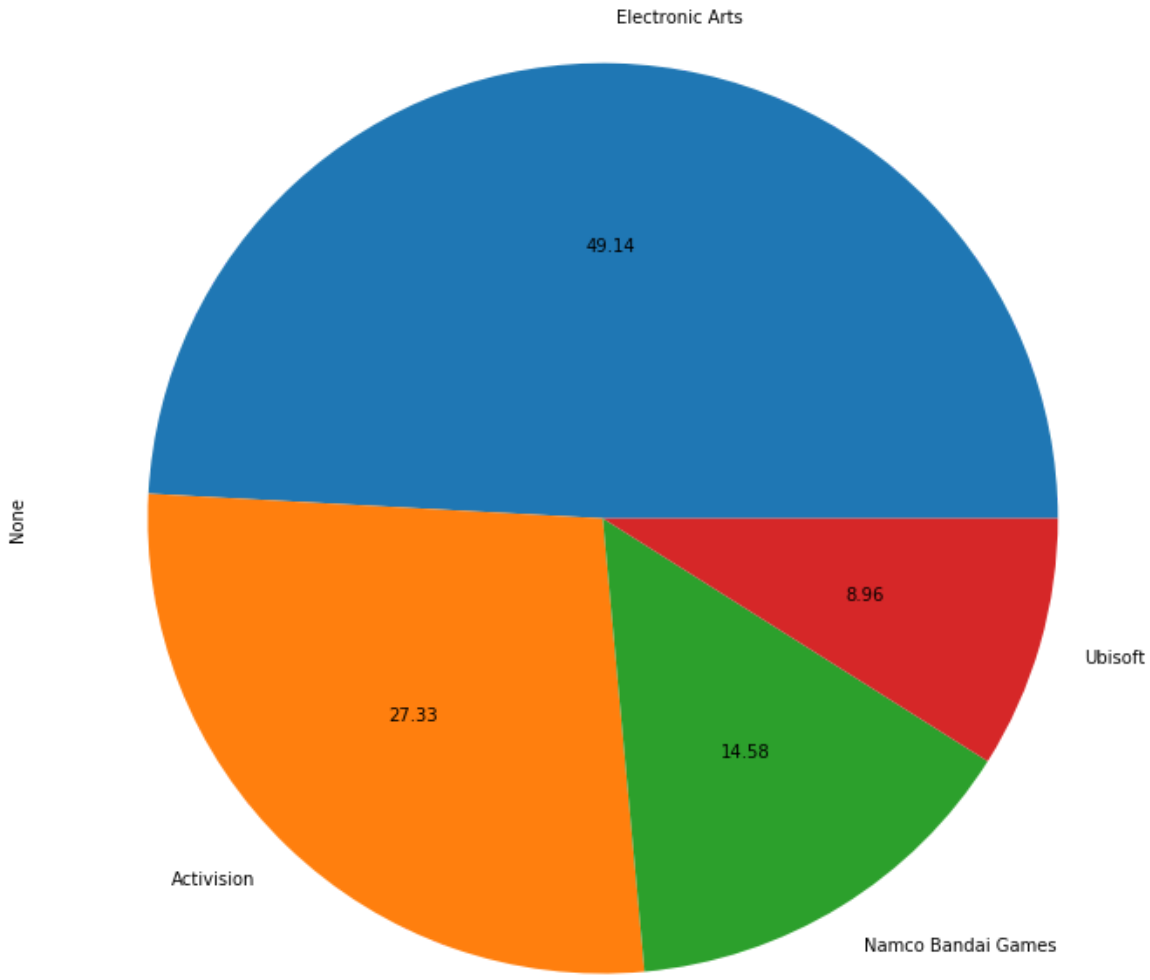
In [33]:

```
print("不同地区的销售额数量饼图")
data_region=data[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']].sum().sort_values(asc
data_region.head(100).plot.pie(labels=label,
                 autopct='%.2f', fontsize=10,figsize=(12, 12))
```

不同地区的销售额数量饼图

Out[33]:

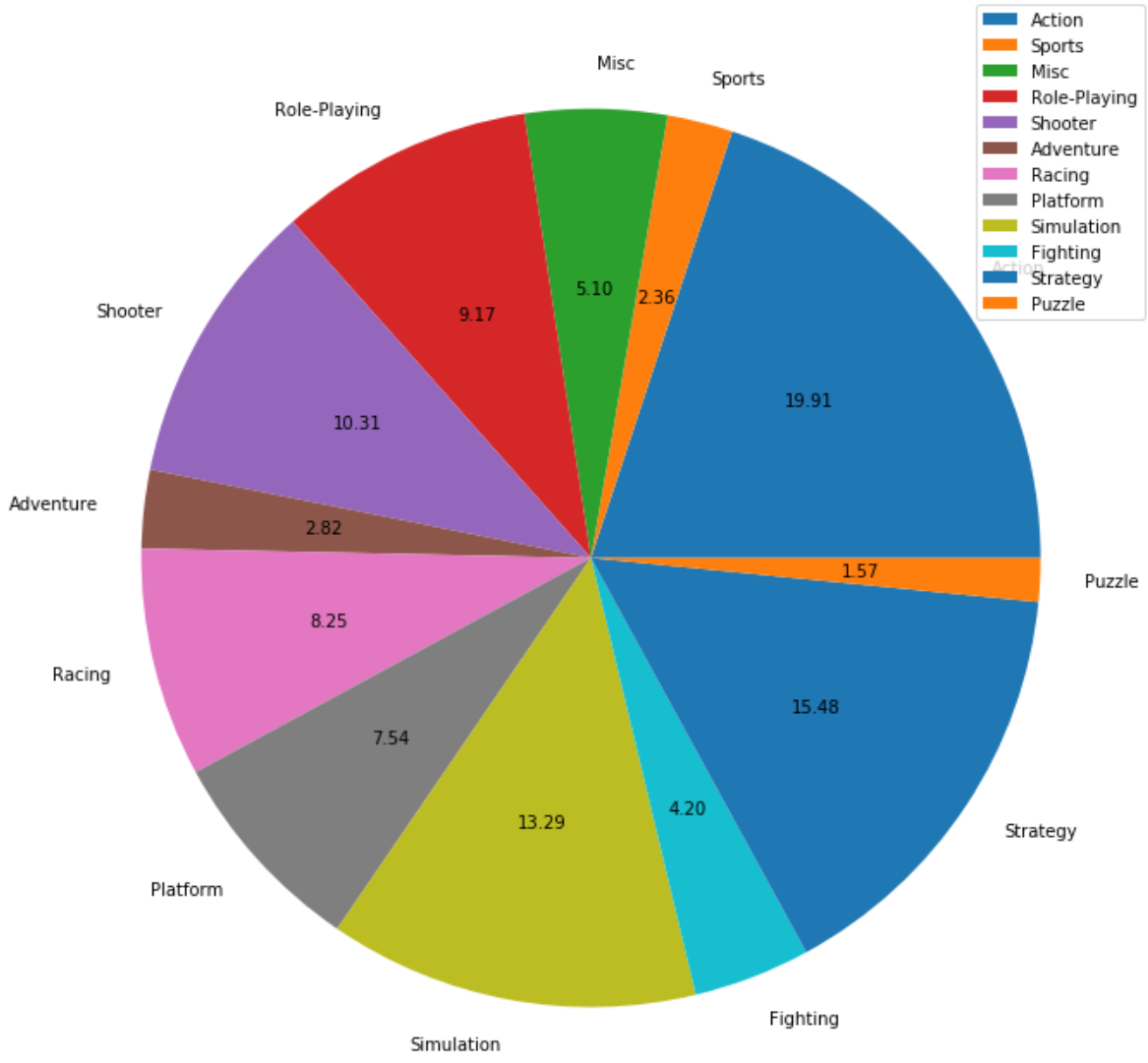
```
<matplotlib.axes._subplots.AxesSubplot at 0x20a12d867b8>
```



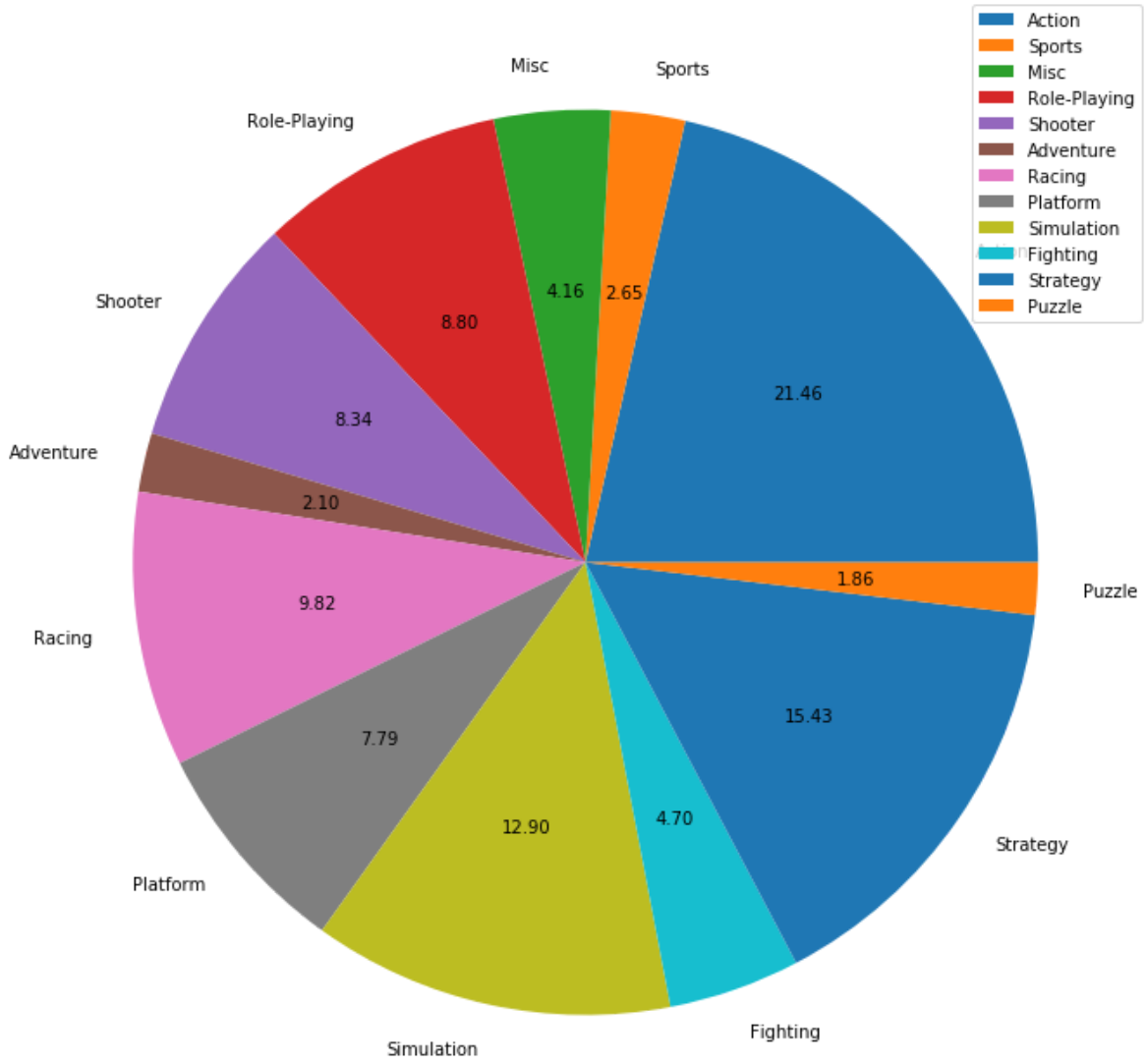
In [41]:

```
location=['NA_Sales',  'EU_Sales' , 'JP_Sales',  'Other_Sales']
for loc in location:
    print(loc,"地区的销售额占比饼图")
    data_loc=data[['Genre',loc ]].groupby('Genre').sum().sort_values(by='Genre')
    label=[]
    for key in data['Genre'].value_counts().index:
        label.append(key)
    data_loc.plot.pie(labels=label,subplots=True,
                      autopct='%.2f', fontsize=10,figsize=(12, 12))
    plt.ylabel('')
    plt.show()
```

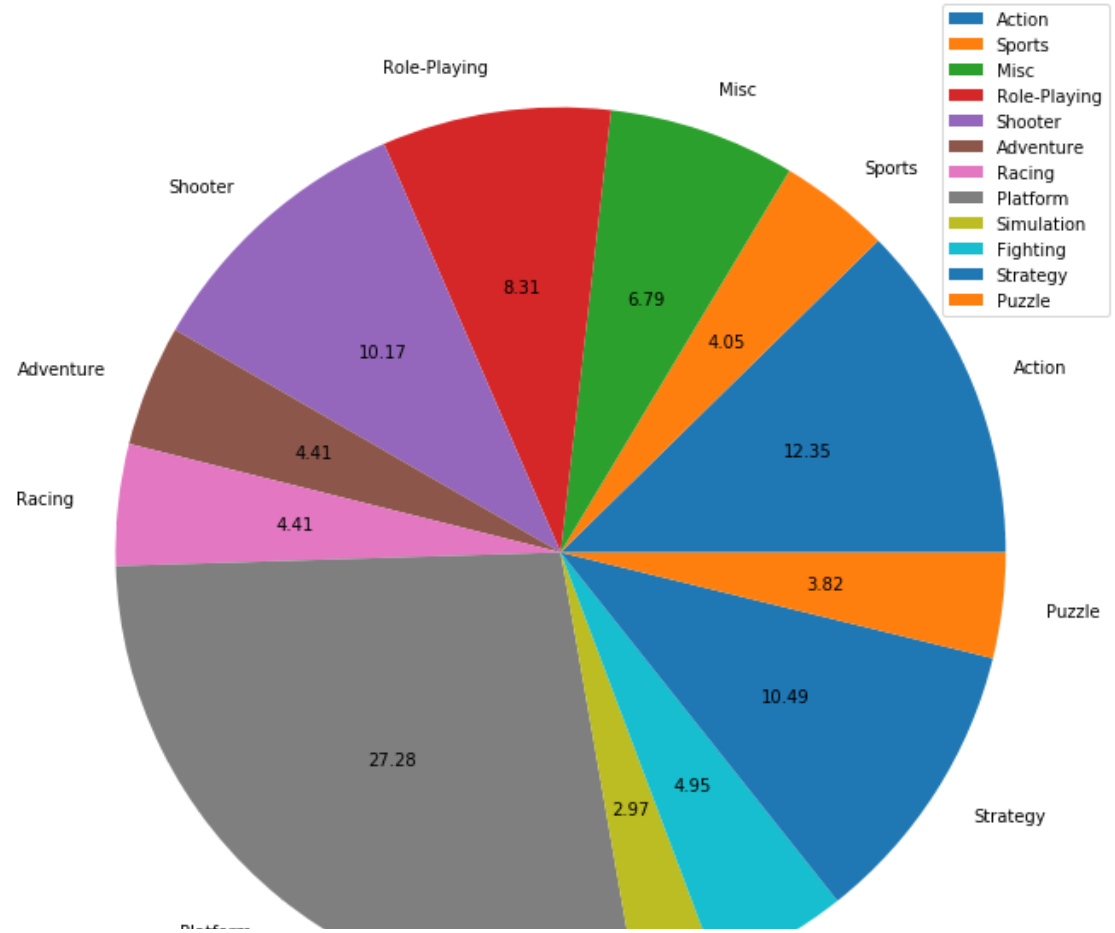
NA_Sales 地区的销售额占比饼图



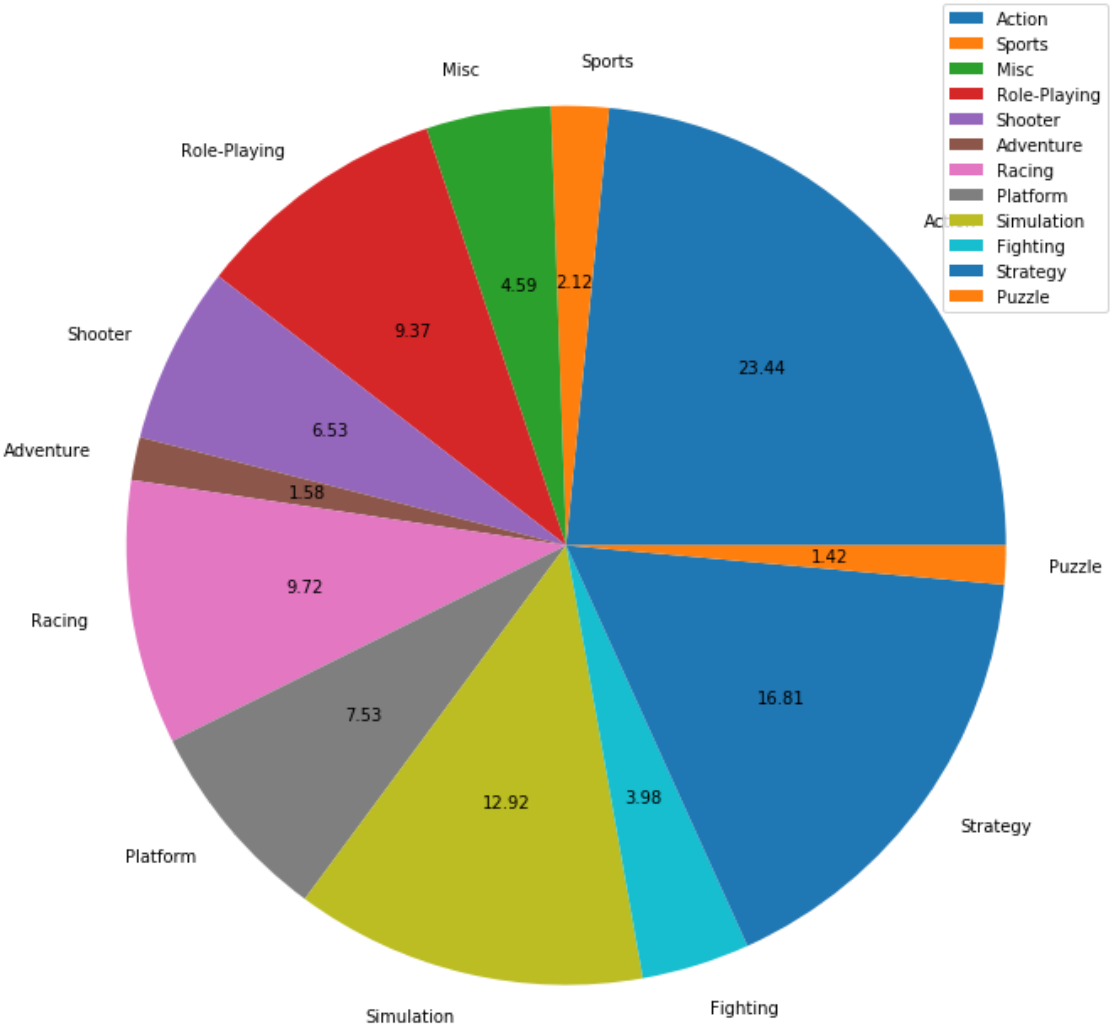
`EU_Sales` 地区的销售额占比饼图



`JP_Sales` 地区的销售额占比饼图



`Other_Sales` 地区的销售额占比饼图


Action
Sports
Misc
Role-Playing
Shooter
Adventure
Racing
Platform
Simulation
Fighting
Strategy
Puzzle
Misc
Sports
Role-Playing
Shooter
Adventure
Racing
Platform
Simulation
Fighting
Strategy
Puzzle
4.59
2.12
9.37
23.44
6.53
1.58
9.72
1.42
7.53
16.81
12.92
3.98

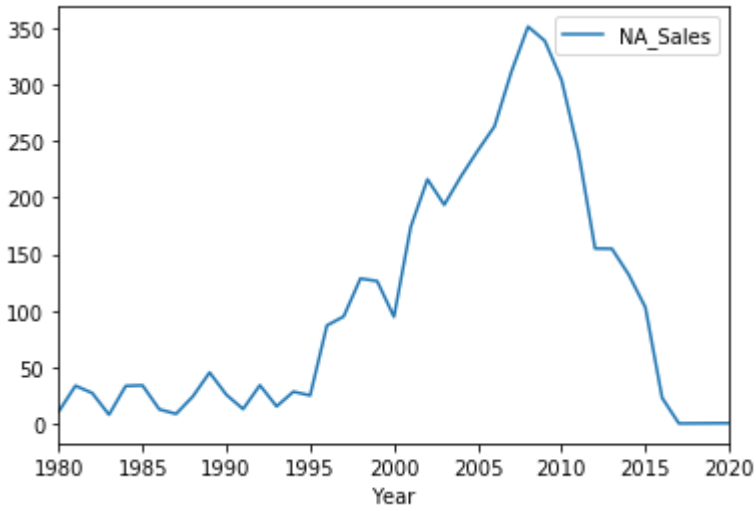
In [47]:

```
print("NA_Sales地区的销售额占比曲线")
data_loc=data[['NA_Sales','Year' ]].groupby('Year').sum().sort_values(by='Year')
data_loc.plot()
```

NA_Sales地区的销售额占比曲线

Out[47]:

<matplotlib.axes._subplots.AxesSubplot at 0x20a18f58a58>



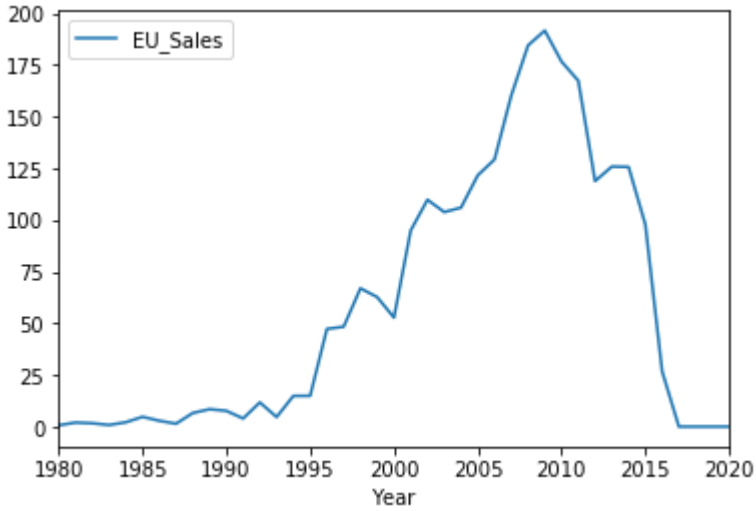
In [48]:

```
print("EU_Sales地区的销售额占比曲线")
data_loc=data[['EU_Sales','Year' ]].groupby('Year').sum().sort_values(by='Year')
data_loc.plot()
```

EU_Sales地区的销售额占比曲线

Out[48]:

<matplotlib.axes._subplots.AxesSubplot at 0x20a18fc5828>



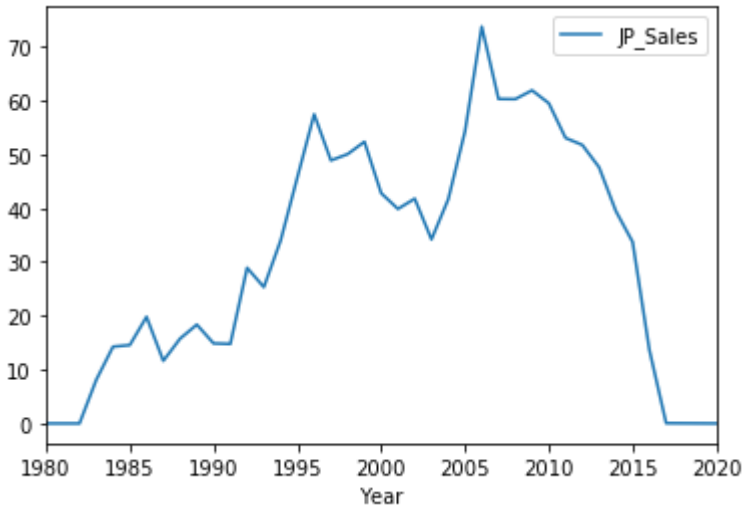
In [49]:

```
print("JP_Sales地区的销售额占比曲线")
data_loc=data[['JP_Sales','Year' ]].groupby('Year').sum().sort_values(by='Year')
data_loc.plot()
```

JP_Sales地区的销售额占比曲线

Out[49]:

<matplotlib.axes._subplots.AxesSubplot at 0x20a1903b5f8>



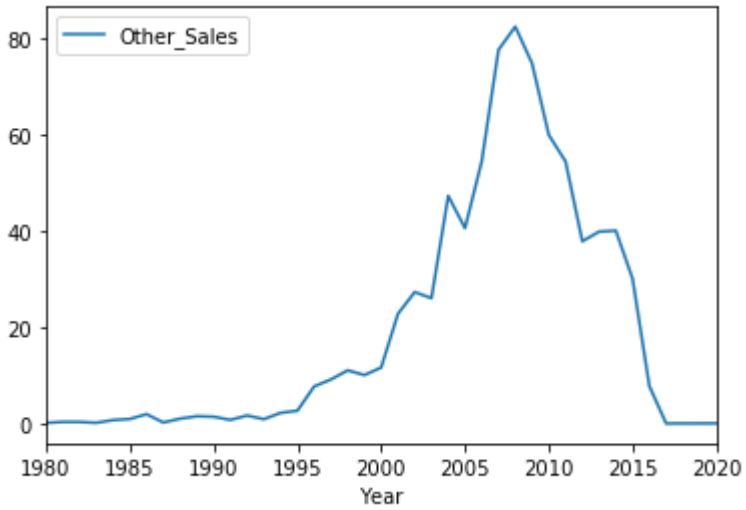
In [50]:

```
print("Other_Sales地区的销售额占比曲线")
data_loc=data[['Other_Sales','Year' ]].groupby('Year').sum().sort_values(by='Year')
data_loc.plot()
```

Other_Sales地区的销售额占比曲线

Out[50]:

<matplotlib.axes._subplots.AxesSubplot at 0x20a18d8b6d8>



In [ ]:

```

```