In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df_players = pd.read_csv("D:\highest_earning_players.csv")
df_teams = pd.read_csv("D:\highest_earning_teams.csv")
df_country = pd.read_csv("D:\country-and-continent-codes-list.csv")
```

In [3]:
```
df_players.head()
```

Out[3]:

	PlayerId	NameFirst	NameLast	CurrentHandle	CountryCode	TotalUSDPrize	Game	Gen
0	3883	Peter	Rasmussen	dupreeh	dk	1822989.41	Counter-Strike: Global Offensive	Fir Pers Shoot
1	3679	Andreas	Højsleth	Xyp9x	dk	1799288.57	Counter-Strike: Global Offensive	Fir Pers Shoot
2	3885	Nicolai	Reedtz	dev1ce	dk	1787489.88	Counter-Strike: Global Offensive	Fir Pers Shoot
3	3672	Lukas	Rossander	gla1ve	dk	1652350.75	Counter-Strike: Global Offensive	Fir Pers Shoot
4	17800	Emil	Reif	Magisk	dk	1416448.64	Counter-Strike: Global Offensive	Fir Pers Shoot

In [4]:
```
df_players.tail()
```

Out[4]:

	PlayerId	NameFirst	NameLast	CurrentHandle	CountryCode	TotalUSDPrize	Game	
995	7400	Janne	Mikkonen	Savjz	fi	50734.44	Hearthstone	C
996	3255	Drew	Biessener	Tidesoftime	us	50449.60	Hearthstone	C
997	49164	Simone	Liguori	Leta	it	49300.00	Hearthstone	C
998	43043	Mike	Eichner	Ike	us	48550.00	Hearthstone	C
999	1100	Jeffrey	Brusi	SjoW	se	47973.61	Hearthstone	C

In [5]:
```
df_players.shape
```

Out[5]:
```
(1000, 8)
```

In [6]:
```
df_players.columns
```

Out[6]:
```
Index(['PlayerId', 'NameFirst', 'NameLast', 'CurrentHandle', 'CountryCode',
       'TotalUSDPrize', 'Game', 'Genre'],
      dtype='object')
```

In [7]:
```
df_players.duplicated().sum()
```

Out[7]:
```
0
```

In [8]:
```
df_players.isnull().sum()
```

Out[8]:
```
PlayerId         0
NameFirst        0
NameLast         0
CurrentHandle    0
CountryCode      0
TotalUSDPrize    0
Game             0
Genre            0
dtype: int64
```

In [9]:
```
df_players.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   PlayerId       1000 non-null   int64  
 1   NameFirst      1000 non-null   object 
 2   NameLast       1000 non-null   object 
 3   CurrentHandle  1000 non-null   object 
 4   CountryCode    1000 non-null   object 
 5   TotalUSDPrize  1000 non-null   float64
 6   Game           1000 non-null   object 
 7   Genre          1000 non-null   object 
dtypes: float64(1), int64(1), object(6)
memory usage: 62.6+ KB
```

In [10]:
```
df_players.describe()
```

Out[10]:

	PlayerId	TotalUSDPrize
count	1000.000000	1.000000e+03
mean	27793.587000	3.977932e+05
std	22170.225194	6.908492e+05
min	1000.000000	2.417167e+04
25%	5374.500000	8.378962e+04
50%	23502.000000	1.683284e+05
75%	48127.250000	3.937352e+05
max	83085.000000	6.952597e+06

In [11]:
```
df_players.nunique()
```

Out[11]:
```
PlayerId         998
NameFirst        756
NameLast         637
CurrentHandle    990
CountryCode       56
TotalUSDPrize    961
Game              10
Genre              5
dtype: int64
```

In [12]:
```
df_teams.head()
```

Out[12]:

	TeamId	TeamName	TotalUSDPrize	TotalTournaments	Game	Genre
0	760	San Francisco Shock	3105000.0	7	Overwatch	First-Person Shooter
1	776	London Spitfire	1591136.5	13	Overwatch	First-Person Shooter
2	768	New York Excelsior	1572618.5	18	Overwatch	First-Person Shooter
3	773	Philadelphia Fusion	1186278.5	15	Overwatch	First-Person Shooter
4	766	Seoul Dynasty	1130000.0	6	Overwatch	First-Person Shooter

In [13]:
```
df_teams.tail()
```

Out[13]:

	TeamId	TeamName	TotalUSDPrize	TotalTournaments	Game	Genre
923	24781	Rex Regum Qeon	6286.8	2	Arena of Valor	Multiplayer Online Battle Arena
924	261	Alliance	4000.0	1	Arena of Valor	Multiplayer Online Battle Arena
925	713	Marines Esports	3429.6	1	Arena of Valor	Multiplayer Online Battle Arena
926	608	British National Team	2500.0	1	Arena of Valor	Multiplayer Online Battle Arena
927	584	Swedish National Team	2500.0	1	Arena of Valor	Multiplayer Online Battle Arena

In [14]:
```
df_teams.shape
```

Out[14]:
```
(928, 6)
```

In [15]:
```
df_teams.columns
```

Out[15]:
```
Index(['TeamId', 'TeamName', 'TotalUSDPrize', 'TotalTournaments', 'Game',
       'Genre'],
      dtype='object')
```

In [16]:
```
df_teams.duplicated().sum()
```

Out[16]:
```
0
```

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

Out[17]:
```
TeamId             0
TeamName           0
TotalUSDPrize      0
TotalTournaments   0
Game               0
Genre              0
dtype: int64
```

In [18]:
```
df_teams.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 928 entries, 0 to 927
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   TeamId            928 non-null    int64  
 1   TeamName          928 non-null    object 
 2   TotalUSDPrize     928 non-null    float64
 3   TotalTournaments  928 non-null    int64  
 4   Game              928 non-null    object 
 5   Genre             928 non-null    object 
dtypes: float64(1), int64(2), object(3)
memory usage: 43.6+ KB
```

In [19]:
```
df_teams.describe()
```

Out[19]:

	TeamId	TotalUSDPrize	TotalTournaments
count	928.000000	9.280000e+02	928.000000
mean	3836.927802	5.399183e+05	31.696121
std	8438.383941	1.902399e+06	61.075848
min	101.000000	1.750000e+02	1.000000
25%	227.000000	3.915000e+04	4.000000
50%	529.000000	1.165306e+05	11.000000
75%	789.000000	3.231491e+05	33.000000
max	24997.000000	3.381064e+07	808.000000

In [20]:
```
df_teams.nunique()
```

Out[20]:
```
TeamId             505
TeamName           505
TotalUSDPrize      854
TotalTournaments   145
Game                10
Genre                5
dtype: int64
```

In [21]:
```
df_country.head()
```

Out[21]:

	Continent_Name	Continent_Code	Country_Name	Two_Letter_Country_Code	Three_Letter_Cou
0	Asia	AS	Afghanistan, Islamic Republic of	AF	
1	Europe	EU	Albania, Republic of	AL	
2	Antarctica	AN	Antarctica (the territory South of 60 deg S)	AQ	
3	Africa	AF	Algeria, People's Democratic Republic of	DZ	
4	Oceania	OC	American Samoa	AS	

In [22]:
```
df_country.tail()
```

Out[22]:

	Continent_Name	Continent_Code	Country_Name	Two_Letter_Country_Code	Three_Letter_C
257	Africa	AF	Zambia, Republic of	ZM	
258	Oceania	OC	Disputed Territory	XX	
259	Asia	AS	Iraq-Saudi Arabia Neutral Zone	XE	
260	Asia	AS	United Nations Neutral Zone	XD	
261	Asia	AS	Spratly Islands	XS	

In [23]:
```
df_country.shape
```

Out[23]:
```
(262, 6)
```

In [24]:
```
df_country.columns
```

Out[24]:
```
Index(['Continent_Name', 'Continent_Code', 'Country_Name',
       'Two_Letter_Country_Code', 'Three_Letter_Country_Code',
       'Country_Number'],
      dtype='object')
```

In [25]:
```
df_country.duplicated().sum()
```

Out[25]:
```
0
```

In [26]:
```
df_country.isnull().sum()
```

Out[26]:
```
Continent_Name               0
Continent_Code              43
Country_Name                 0
Two_Letter_Country_Code      1
Three_Letter_Country_Code    4
Country_Number               4
dtype: int64
```

In [27]:
```
df_country.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 262 entries, 0 to 261
Data columns (total 6 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Continent_Name            262 non-null    object
 1   Continent_Code            219 non-null    object
 2   Country_Name              262 non-null    object
 3   Two_Letter_Country_Code   261 non-null    object
 4   Three_Letter_Country_Code 258 non-null    object
 5   Country_Number            258 non-null    float64
dtypes: float64(1), object(5)
memory usage: 12.4+ KB
```

In [28]:
```
df_teams.describe()
```

Out[28]:

	TeamId	TotalUSDPrize	TotalTournaments
count	928.000000	9.280000e+02	928.000000
mean	3836.927802	5.399183e+05	31.696121
std	8438.383941	1.902399e+06	61.075848
min	101.000000	1.750000e+02	1.000000
25%	227.000000	3.915000e+04	4.000000
50%	529.000000	1.165306e+05	11.000000
75%	789.000000	3.231491e+05	33.000000
max	24997.000000	3.381064e+07	808.000000

In [29]:
```
df_country.nunique()
```

Out[29]:
```
Continent_Name              7
Continent_Code              6
Country_Name              254
Two_Letter_Country_Code   253
Three_Letter_Country_Code 250
Country_Number            250
dtype: int64
```

In [30]:
```
print(df_players['Game'].value_counts())
```

```
Counter-Strike: Global Offensive    100
Dota 2                              100
League of Legends                   100
Fortnite                            100
Overwatch                           100
Starcraft II                        100
Heroes of the Storm                 100
PUBG                                100
Arena of Valor                      100
Hearthstone                         100
Name: Game, dtype: int64
```

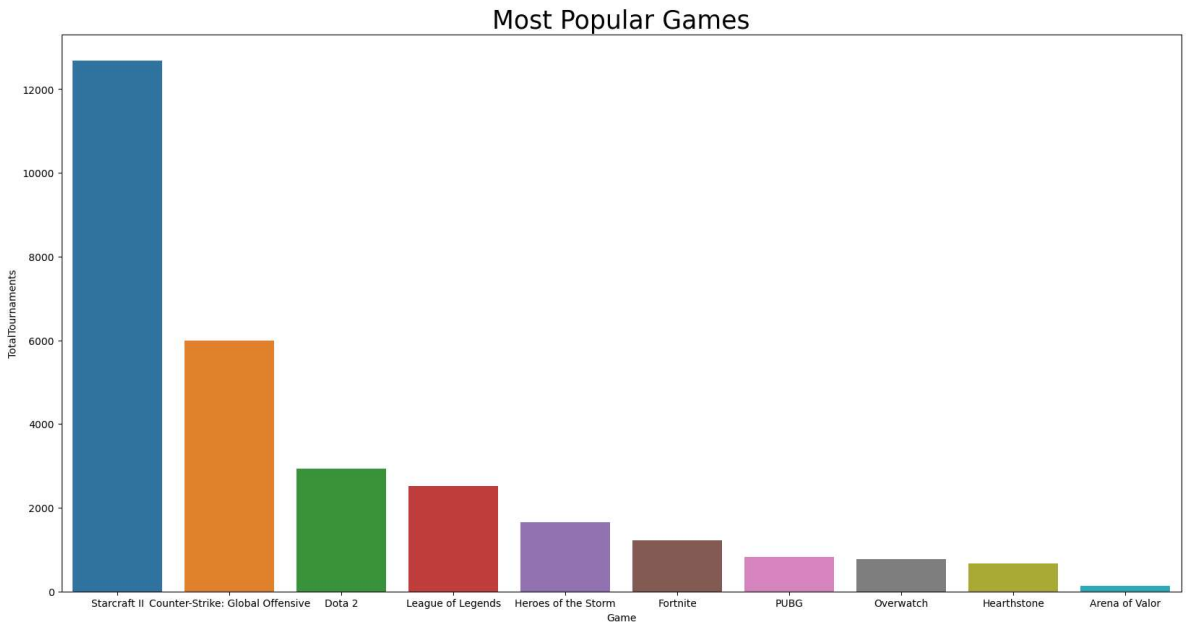
In [31]:
```
most_popular = pd.DataFrame(df_teams.groupby('Game')['TotalTournaments'].sum()
most_popular
```

Out[31]:

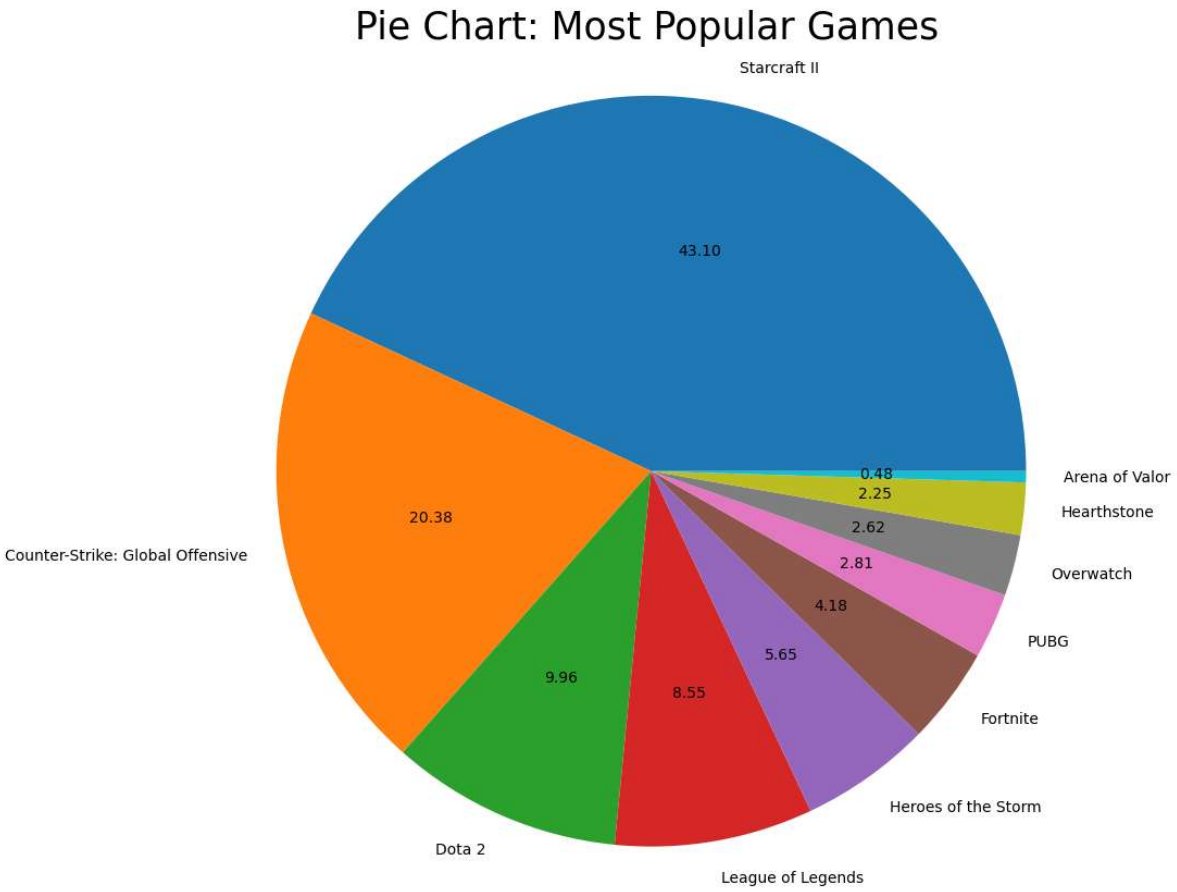
	Game	TotalTournaments
0	Starcraft II	12676
1	Counter-Strike: Global Offensive	5996
2	Dota 2	2931
3	League of Legends	2515
4	Heroes of the Storm	1663
5	Fortnite	1229
6	PUBG	828
7	Overwatch	772
8	Hearthstone	662
9	Arena of Valor	142

In [32]:
```
plt.figure(figsize=(20,10))
sns.barplot(x=most_popular['Game'], y=most_popular['TotalTournaments'])
plt.title('Most Popular Games', size=25)
```

Out[32]:
```
Text(0.5, 1.0, 'Most Popular Games')
```



In [33]:
```
plt.figure(figsize=(12,10))
plt.pie(most_popular['TotalTournaments'], labels=most_popular['Game'], autopct
plt.title('Pie Chart: Most Popular Games', size=25)
plt.axis('equal')
plt.show()
```



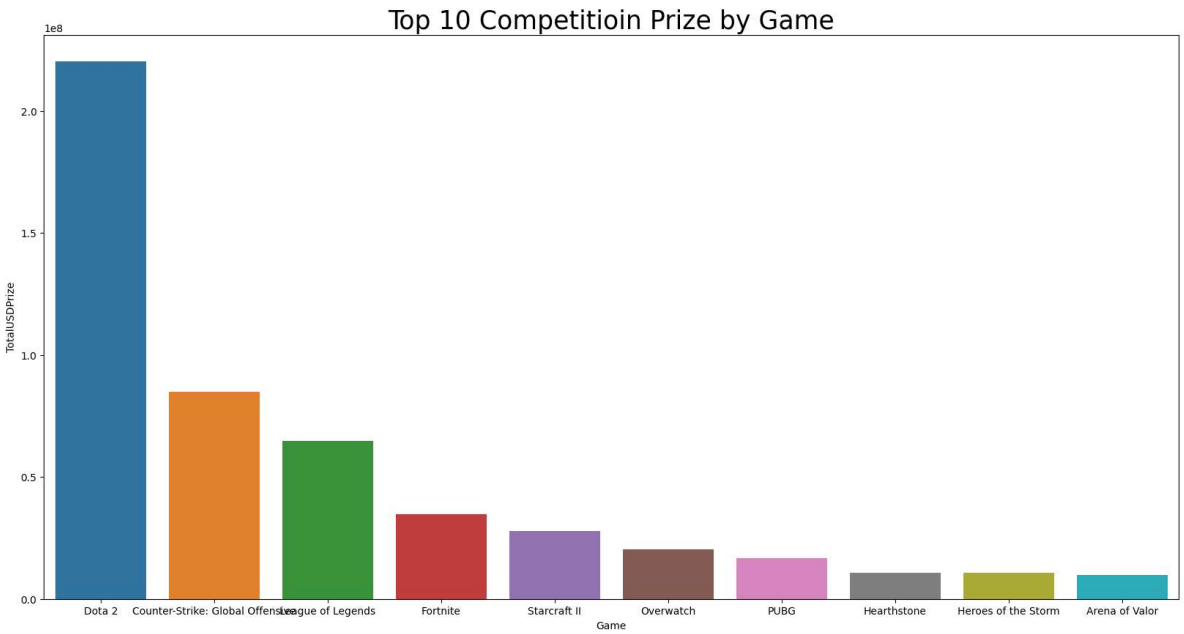
In [34]:
```
most_prize = pd.DataFrame(df_teams.groupby('Game')['TotalUSDPrize'].sum().sort
most_prize
```

Out[34]:

	Game	TotalUSDPrize
0	Dota 2	2.202828e+08
1	Counter-Strike: Global Offensive	8.485393e+07
2	League of Legends	6.466556e+07
3	Fortnite	3.466133e+07
4	Starcraft II	2.785615e+07
5	Overwatch	2.046527e+07
6	PUBG	1.671500e+07
7	Hearthstone	1.086453e+07
8	Heroes of the Storm	1.071052e+07
9	Arena of Valor	9.969149e+06

In [35]:
```
plt.figure(figsize=(20,10))
sns.barplot(x=most_prize['Game'], y=most_prize['TotalUSDPrize'])
plt.title('Top 10 Competitioin Prize by Game', size=25)
```

Out[35]:
```
Text(0.5, 1.0, 'Top 10 Competitioin Prize by Game')
```



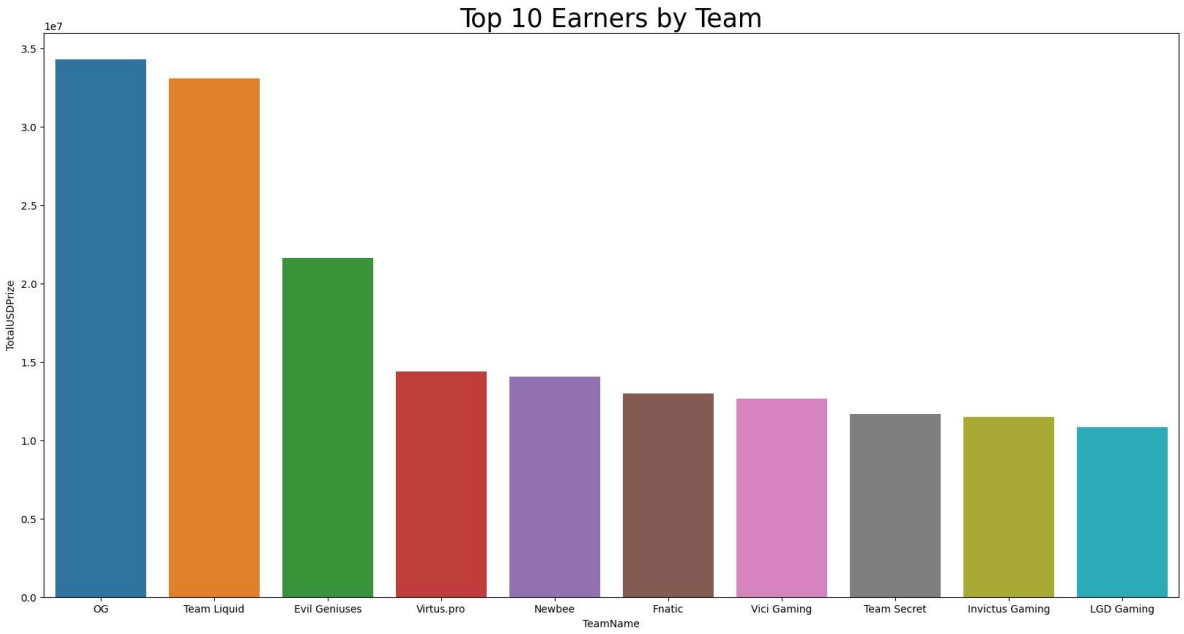
In [36]:
```
top_10_team = pd.DataFrame(df_teams.groupby('TeamName')['TotalUSDPrize'].sum()
top_10_team
```

Out[36]:

	TeamName	TotalUSDPrize
0	OG	34297886.13
1	Team Liquid	33095692.87
2	Evil Geniuses	21662171.52
3	Virtus.pro	14393878.63
4	Newbee	14072159.40
5	Fnatic	13000709.75
6	Vici Gaming	12660736.30
7	Team Secret	11688870.47
8	Invictus Gaming	11515644.56
9	LGD Gaming	10852395.33

In [37]:
```
plt.figure(figsize=(20,10))
sns.barplot(x=top_10_team['TeamName'], y=top_10_team['TotalUSDPrize'])
plt.title('Top 10 Earners by Team', size=25)
```

Out[37]:
```
Text(0.5, 1.0, 'Top 10 Earners by Team')
```



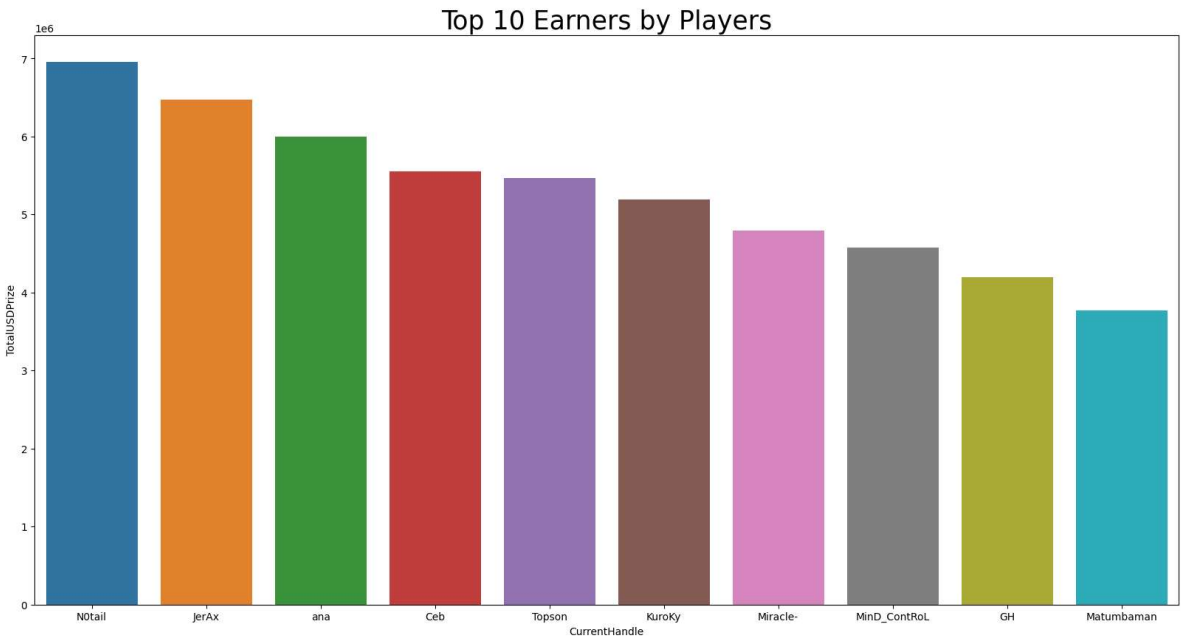
In [38]:
```
top_10_player = df_players[['NameFirst', 'NameLast', 'CurrentHandle', 'TotalUS
top_10_player
```

Out[38]:

	NameFirst	NameLast	CurrentHandle	TotalUSDPrize
0	Johan	Sundstein	N0tail	6952596.58
1	Jesse	Vainikka	JerAx	6470000.02
2	Anathan	Pham	ana	6000411.96
3	Sébastien	Debs	Ceb	5554297.41
4	Topias	Taavitsainen	Topson	5470902.57
5	Kuro	Takhasomi	KuroKy	5193382.81
6	Amer	Al-Barkawi	Miracle-	4798043.68
7	Ivan	Ivanov	MinD_ContRoL	4579118.16
8	Maroun	Merhej	GH	4193412.69
9	Lasse	Urpalainen	Matumbaman	3765369.04

In [39]:
```
plt.figure(figsize=(20,10))
sns.barplot(x=top_10_player['CurrentHandle'], y=top_10_player['TotalUSDPrize']
plt.title('Top 10 Earners by Players', size=25)
```

Out[39]:
```
Text(0.5, 1.0, 'Top 10 Earners by Players')
```



In [40]:
```
from sklearn.linear_model import LinearRegression
```

In [41]:
```
lm = LinearRegression()
lm
```

Out[41]:
```
LinearRegression()
```

In [42]:
```
x = df_teams[['TotalTournaments']]
y = df_teams[['TotalUSDPrize']]
lm.fit(x,y)
```

Out[42]:
```
LinearRegression()
```

In [43]:
```
Yhat = lm.predict(x)
Yhat[0:5]
```

Out[43]:
```
array([[388332.90962656],
       [425161.06141115],
       [455851.18789831],
       [437437.11200602],
       [382194.88432913]])
```

In [44]:
```
lm.intercept_
```

Out[44]:
```
array([345366.73254453])
```

In [45]:
```
lm.coef_
```

Out[45]:
```
array([[6138.02529743]])
```

In [46]:
```
# total earning =yhat=a+bx= [345366.73254453+6138.02529743] no. of tournaments
```

In [47]:
```
# correlation matrix
```

In [48]:
```
pd_teams = pd.read_csv("D:\highest_earning_teams.csv")
```

In [49]:
```
pd_teams.corr()
```

Out[49]:

	TeamId	TotalUSDPrize	TotalTournaments
TeamId	1.000000	-0.076652	-0.139735
TotalUSDPrize	-0.076652	1.000000	0.197059
TotalTournaments	-0.139735	0.197059	1.000000

In [50]:
```
df_numerized = pd_teams

for col_name in df_numerized.columns:
    if(df_numerized[col_name].dtype == 'object'):
        df_numerized[col_name] = df_numerized[col_name].astype('category')
        df_numerized[col_name] = df_numerized[col_name].cat.codes

df_numerized
```

Out[50]:

	TeamId	TeamName	TotalUSDPrize	TotalTournaments	Game	Genre
0	760	334	3105000.0	7	7	2
1	776	208	1591136.5	13	7	2
2	768	247	1572618.5	18	7	2
3	773	286	1186278.5	15	7	2
4	766	339	1130000.0	6	7	2
...	...	...	...	...	...	...
923	24781	316	6286.8	2	0	3
924	261	18	4000.0	1	0	3
925	713	226	3429.6	1	0	3
926	608	53	2500.0	1	0	3
927	584	362	2500.0	1	0	3

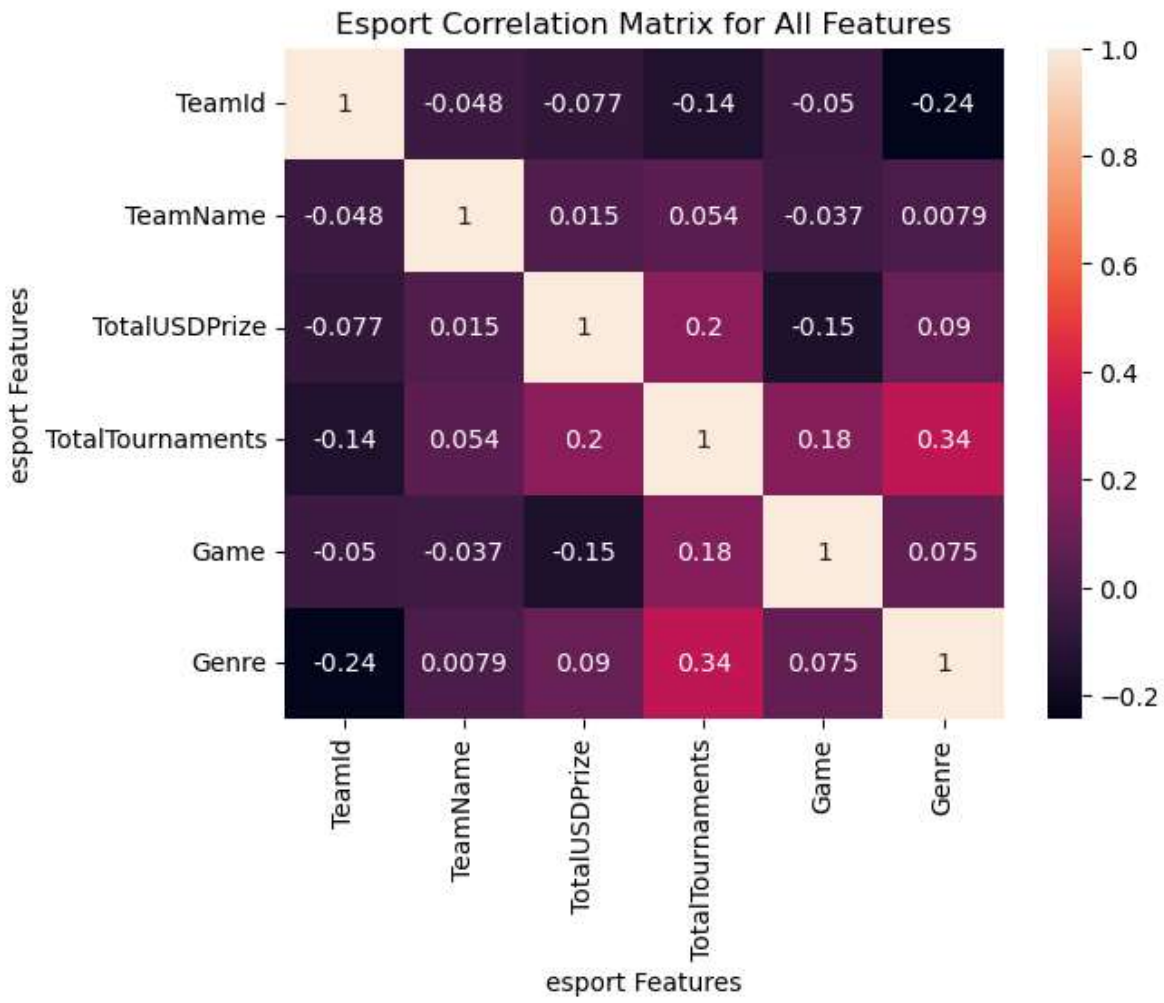
928 rows × 6 columns

In [51]:
```
correlation_matrix = df_numerized.corr(method='pearson')

sns.heatmap(correlation_matrix, annot=True)

plt.title('Esport Correlation Matrix for All Features')
plt.xlabel('esport Features')
plt.ylabel('esport Features')
```

Out[51]: Text(50.72222222222221, 0.5, 'esport Features')



In [ ]: