

# flashpoint\_analytics\_python

April 24, 2023

## 1 Flashpoint Analytics

[Flashpoint](#) is a web-game preservation project, made in 2018 in an effort to save as many games as possible from the then upcoming Flash End-Of-Life, while also making them playable for everyone. Today, it hosts more than 170 000 games and thousands of active users all around the world.

This notebook contains a descriptive statistical analysis about the games available in Flashpoint, with an emphasis on categorical data, such as the technology that was used to make them or the publisher who used to host them in the past.

The Flashpoint database, which keeps all the data that will be used in the analysis, can be found [here](#).

### 1.1 Import modules

```
[1]: import os
import urllib.request
import sqlite3
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
```

These are the modules that we are going to use for our analysis. Notably: \* *sqlite3* allows us to communicate with a SQLite database; \* *pandas* offers very useful tools for working with data; \* *seaborn* and *matplotlib* come with some handful functions to display and visualize data.

### 1.2 Retrieve data

```
[2]: os.mkdir("data")

# download data from the source

url = "http://infinity.unstable.life/Flashpoint/Data/flashpoint.sqlite"
filename = "data/flashpoint.sqlite"
urllib.request.urlretrieve(url, filename)

# connect to the database and store the "game" table in a dataframe
```

```
con = sqlite3.connect("data/flashpoint.sqlite")
df = pd.read_sql_query("SELECT * FROM game", con)
con.close()
```

### 1.3 Explore data

Let's have a first look at our data.

```
[3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 177469 entries, 0 to 177468
Data columns (total 27 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    177469 non-null object
1   parentGameId         57600 non-null object
2   title                177469 non-null object
3   alternateTitles      177469 non-null object
4   series               177469 non-null object
5   developer            177469 non-null object
6   publisher            177469 non-null object
7   dateAdded            177469 non-null object
8   dateModified         177469 non-null object
9   platform             177469 non-null object
10  broken               177469 non-null int64
11  extreme              177469 non-null int64
12  playMode             177469 non-null object
13  status               177469 non-null object
14  notes                177469 non-null object
15  source               177469 non-null object
16  applicationPath      177469 non-null object
17  launchCommand        177469 non-null object
18  releaseDate          177469 non-null object
19  version              177469 non-null object
20  originalDescription  177469 non-null object
21  language             177469 non-null object
22  library              177469 non-null object
23  orderTitle           177469 non-null object
24  activeDataId         153952 non-null float64
25  activeDataOnDisk     177469 non-null int64
26  tagsStr              177469 non-null object
dtypes: float64(1), int64(3), object(23)
memory usage: 36.6+ MB
```

There is a total of 27 variables and almost all of them belong to the *object* data type. We are going to need only some of them, so let's keep only the relevant ones.

```
[4]: vars_to_keep = ["id", "title", "developer", "publisher", "platform",
    ↪ "releaseDate", "language", "library", "tagsStr"]
df = df[vars_to_keep]
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 177469 entries, 0 to 177468
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    177469 non-null object
1   title                 177469 non-null object
2   developer             177469 non-null object
3   publisher             177469 non-null object
4   platform              177469 non-null object
5   releaseDate          177469 non-null object
6   language              177469 non-null object
7   library               177469 non-null object
8   tagsStr              177469 non-null object
dtypes: object(9)
memory usage: 12.2+ MB
```

Among the remaining variables there are no missing values.

To complete our preliminary analysis, let's print the first rows of our dataframe.

```
[5]: df.head()
```

```
[5]:
```

	id	title \
0	6db72888-6aa5-34c9-0ff3-ffe4cfe0fc61	All Grown Up: Krazy Karts
1	3bba3af6-8e76-b2c8-b423-2d2d8bdfdd50	Showdown: The Gunfighting Game
2	fb479276-2325-4dbb-bafd-64fcc8aeb684	Hamsterball Bowling
3	7cdf5f4-11cb-b1de-51bc-bca13ef78adb	Dunk Tank
4	1d9ff021-2404-9785-bb7d-0fcda67a55d	Baby Knight

	developer	publisher	platform	releaseDate	language	library \
0	Ezone	Nickelodeon	3D Groove GX			arcade
1	3D Groove	3D Groove	3D Groove GX			arcade
2	Ezone	atv.Disney.go.com	3D Groove GX			arcade
3	Ezone	atv.Disney.go.com	3D Groove GX			arcade
4	Pepworks	Pepworks	3D Groove GX			arcade

	tagsStr
0	Racing
1	Shooter
2	Arcade
3	Arcade
4	Platformer

## 1.4 Analyze data

### 1.4.1 Developers and Publishers

It would be interesting to know which are the most prolific developers and publishers. Let's find out by creating a frequency table for each variable and looking at the first ten entries.

```
[6]: top_developers = df["developer"].value_counts()[:10]
      top_developers
```

```
[6]:
      46151
123Bee      2730
Games2Rule  2441
Games2Jolly.com  1915
Selfdefiant 1603
WowEscape.com 1436
Top10NewGames 1144
PalmarianFire 1059
Ena Game Studio 1031
Neopets      957
Name: developer, dtype: int64
```

The first row is blank because some games (actually, most of them) do not have a developer value associated in the database. Let's filter out those entries.

```
[7]: top_developers = df.loc[df.developer != '', "developer"].value_counts()[:10]
      top_developers
```

```
[7]: 123Bee      2730
      Games2Rule  2441
      Games2Jolly.com  1915
      Selfdefiant 1603
      WowEscape.com 1436
      Top10NewGames 1144
      PalmarianFire 1059
      Ena Game Studio 1031
      Neopets      957
      Mirchi Games   812
      Name: developer, dtype: int64
```

These are the most represented developers in the database. It is impressive to notice that almost all of them are known to be specialised in escape games, which we can suppose to be a very popular genre; we will dig into this later. Lastly, a special mention for Neopets, which managed to build a passionate community still active after over 20 years.

```
[8]: top_publishers = df.loc[df.publisher != '', "publisher"].value_counts()[:10]
      top_publishers
```

```
[8]: DeviantArt      7655
      Newgrounds    6550
```

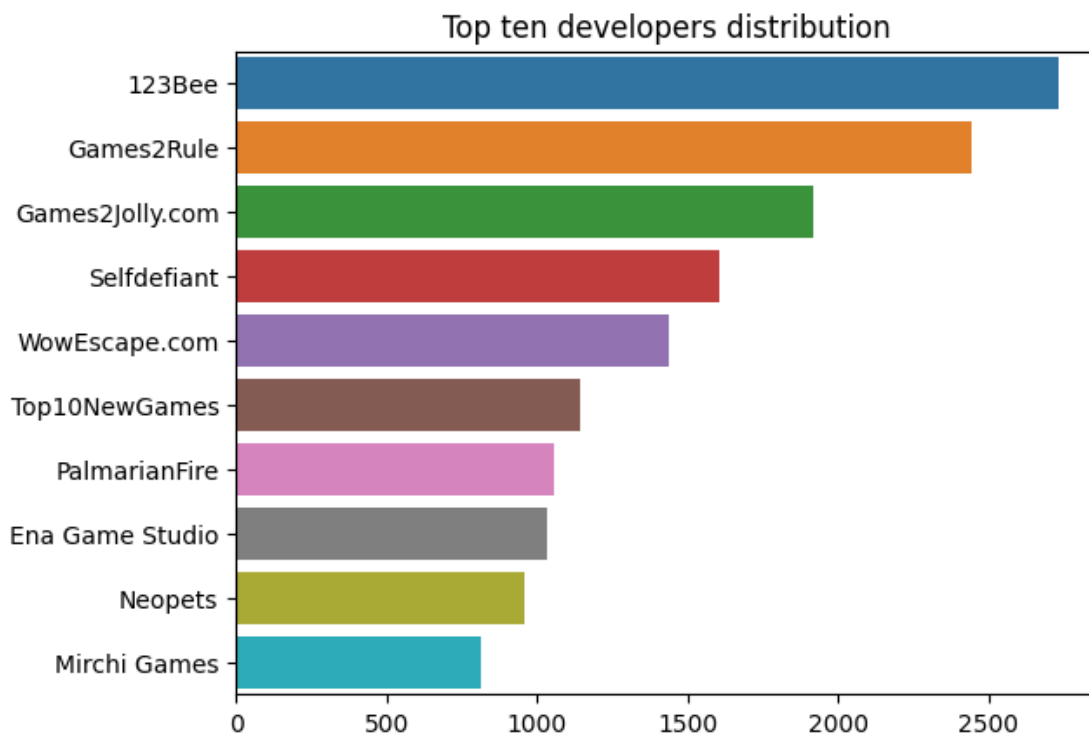
Disney	2428
Nickelodeon	1659
Armor Games	1437
GameMonetize	1374
Eka's Portal	1261
Cartoon Network	1243
Kongregate	1219
Melting-Mindz	1214

Name: publisher, dtype: int64

Among the publishers, we can see some very renowned names, at least in the gaming community, like *Newgrounds*, *Armor Games* and *Kongregate*. There is also a considerable amount of games published by TV broadcasters, such as *Disney*, *Nickelodeon* and *Cartoon Network*, supposedly to promote their shows.

Now let's look at a visualization of the same data, by making use of bar plots and pie charts.

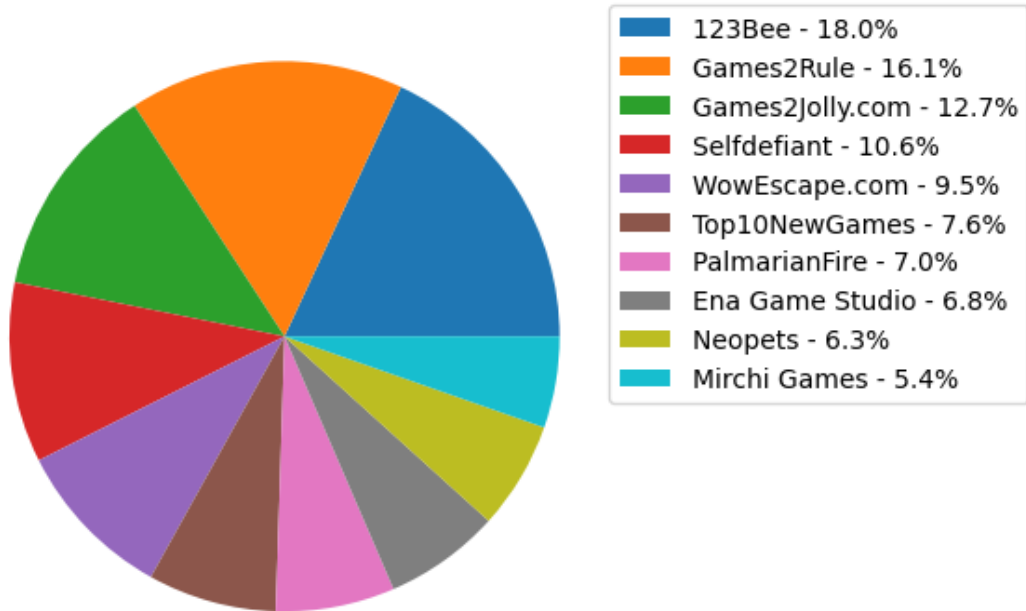
```
[9]: sns.barplot(x = top_developers.values, y = top_developers.index, orient = "h").
      ↪set(title = "Top ten developers distribution");
```



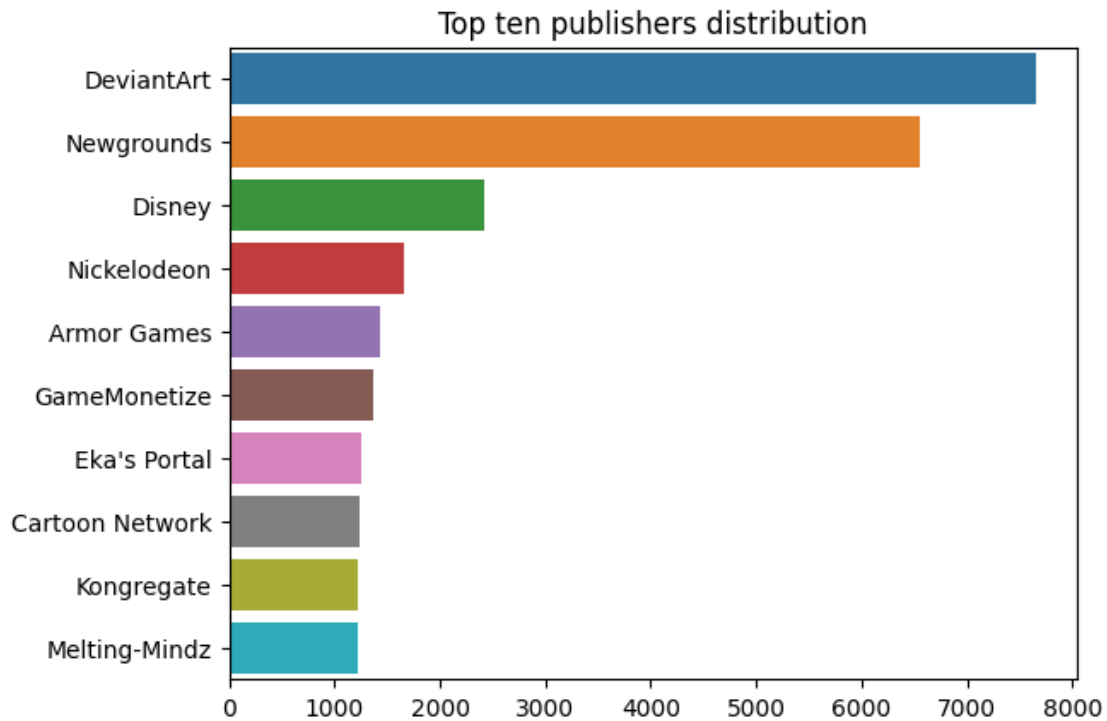
```
[10]: labels = top_developers.index
      sizes = top_developers.values / top_developers.values.sum() * 100
      plt.pie(sizes, textprops = {"color":"w"})
      labels = [f"{l} - {s:0.1f}%" for l, s in zip(labels, sizes)]
```

```
plt.legend(labels = labels, bbox_to_anchor = (1.6,1), loc = "best")
plt.title("Top ten developers distribution")
plt.show()
```

Top ten developers distribution

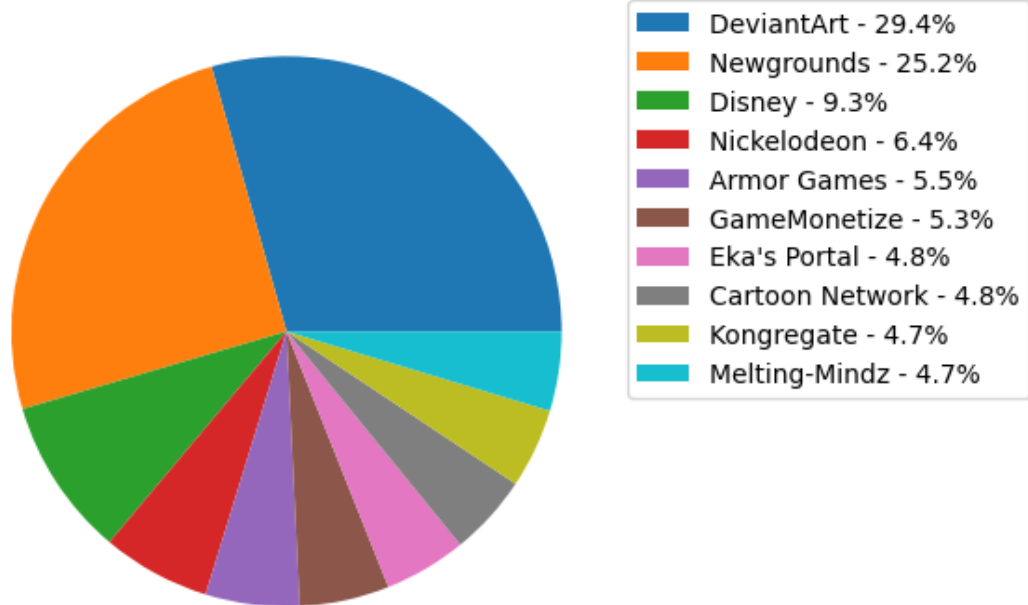


```
[11]: sns.barplot(x = top_publishers.values, y = top_publishers.index, orient = "h").
      ↪ set(title = "Top ten publishers distribution");
```



```
[12]: labels = top_publishers.index
      sizes = top_publishers.values / top_publishers.values.sum() * 100
      plt.pie(sizes, textprops = {"color":"w"})
      labels = [f"{l} - {s:0.1f}%" for l, s in zip(labels, sizes)]
      plt.legend(labels = labels, bbox_to_anchor = (1.6,1), loc = "best")
      plt.title("Top ten publishers distribution")
      plt.show()
```

Top ten publishers distribution



### 1.4.2 Release Dates and Platforms

Flash games started to appear towards the end of the twentieth century and became popular in the next decade. Let's observe the release dates we have got here, being aware that they are not specified for all games.

```
[13]: df_dates = df.loc[(df.releaseDate != ""), ["title", "releaseDate", "platform", "library"]].sort_values(by=["releaseDate"])
df_dates
```

```
[13]:
```

	title	releaseDate	platform	library
26897	TankTrouble	16-12-2007	Flash	arcade
47769	Blastar	1984	HTML5	arcade
141921	Idle Johnny	1993	Shockwave	theatre
155542	ZZZ...I want to sleep	1994	Shockwave	theatre
155541	The Health Checkup	1994	Shockwave	theatre
...	...	...	...	...
37388	GWL Hayley Footjob (Commission)	2080-10-05	Flash	arcade
50821	Havok Xtra Marble Demo	21/08/2001	Shockwave	arcade
50893	Ray Cast Car	27/06/2001	Shockwave	arcade
105547	(Gift) Luna's Christmas gift	2917-12-26	Flash	theatre
50070	Havok Xtra RC Car Demo	7/1/2002	Shockwave	arcade



[75997 rows x 4 columns]

There seems to be a problem with the data. Entries should follow the “YY-MM-DD” date format as per Flashpoint guidelines, but some games come in a different one. In addition, if the exact day or month of release is unknown, it is allowed to specify the year only. Let’s clean up our data for consistency.

```
[14]: warnings.filterwarnings("ignore", category = UserWarning)
df_dates["releaseDate"] = pd.to_datetime(df_dates["releaseDate"], errors = "coerce")
df_dates = df_dates.dropna().sort_values("releaseDate")
df_dates
```

```
[14]:
```

	title	releaseDate	platform	\
47769	Blastar	1984-01-01	HTML5	
141921	Idle Johnny	1993-01-01	Shockwave	
155542	ZZZ...I want to sleep	1994-01-01	Shockwave	
155541	The Health Checkup	1994-01-01	Shockwave	
141942	QP-Shot 1000	1994-01-01	Shockwave	
...	...	...	...	
175851	Awesome Game	2022-12-23	HTML5	
175933	Christmas in Vienna	2022-12-23	HTML5	
175934	Chrysler Building	2022-12-23	HTML5	
177404	T-Mobile Tuesdays: Win \$2,300 for 2023!	2022-12-27	HTML5	
37388	GWL Hayley Footjob (Commission)	2080-10-05	Flash	

	library
47769	arcade
141921	theatre
155542	theatre
155541	theatre
141942	arcade
...	...
175851	arcade
175933	arcade
175934	arcade
177404	arcade
37388	arcade

[75989 rows x 4 columns]

*Pandas* automatically assigned January 1 as month and day for those games whose only date information was the year. There is still one odd observation, the last one, which is most likely a typo.

```
[15]: df_dates = df_dates[:-1]
df_dates[:20]
```

```
[15]:
```

	title	releaseDate	platform	library
47769	Blastar	1984-01-01	HTML5	arcade
141921	Idle Johnny	1993-01-01	Shockwave	theatre
155542	ZZZ...I want to sleep	1994-01-01	Shockwave	theatre
155541	The Health Checkup	1994-01-01	Shockwave	theatre
141942	QP-Shot 1000	1994-01-01	Shockwave	arcade
155545	Dangerous Two	1994-01-01	Shockwave	theatre
73010	Virtual Banana Original	1994-02-01	VRML	arcade
134268	Virtual University of Auckland	1994-02-01	Hyper-G	arcade
102651	Clock Tower	1994-11-17	VRML	arcade
134765	The Austrian National Library	1994-11-17	VRML	arcade
102083	Model of the IICM institute	1994-11-17	VRML	arcade
102104	Office	1994-11-17	VRML	arcade
100915	Office	1994-11-17	Hyper-G	arcade
134763	Graz 3D Model	1994-11-17	VRML	arcade
102674	Engine	1994-11-17	VRML	arcade
102663	Cubes	1994-11-17	VRML	arcade
86424	Clock	1994-11-17	Hyper-G	arcade
134764	Fancy Cubes	1994-11-17	Hyper-G	arcade
102657	Corvette	1994-11-17	VRML	arcade
75234	Missile Commando	1995-01-01	Java	arcade

Finally, we have got our correct release dates. We see that the oldest game in the list is *Blastar*, which was released in 1984. Actually, the game present in Flashpoint is a HTML5 version, which was developed and released much more recently. Moving on, starting from 1993 we recognize some old technologies, such as *Shochwave*, *VRML* and *Hyper-G*.

We can actually distinguish between proper games and animations by looking at the *library* column: the former are labeled with *arcade*, the latter with *theatre* values. Thus, the oldest animation featured is *Idle Johnny* from 1993, while the first “true” game (not counting *Blastar*) could be either *QP-Shot 1000* (which came out at some time in 1994), or *Virtual Banana Original* and *Virtual University of Auckland*, both from February 1st, 1994.

```
[16]: df_dates = df_dates[1:]
df_dates[-20:]
```

```
[16]:
```

	title	releaseDate	\
175058	Um Conto da Cloe	2022-12-08	
176260	Pad of Time	2022-12-08	
175153	/f/ 101	2022-12-09	
176537	Vault of the Pineapples	2022-12-09	
172001	Snowsgiving Choose Your Own Adventure	2022-12-10	
177431	look_into_my_eyes_its_open_wide.swf	2022-12-10	
176112	In the Name of Freedom: Black Apocalypse	2022-12-13	
175448	Mcdonalds Final	2022-12-13	
172414	The Smurfs Cooking	2022-12-13	
172155	Cartoon Network Winter Games	2022-12-14	
176208	Monkey Mart	2022-12-14	

176171	Maptroid: Worlds	2022-12-16
176557	Why is This a Curation?	2022-12-17
176901	Monster Bar Simulator	2022-12-18
177091	Wubbzy explains facts about Mars.	2022-12-19
176747	Defender of Ukraine	2022-12-20
175851	Awesome Game	2022-12-23
175933	Christmas in Vienna	2022-12-23
175934	Chrysler Building	2022-12-23
177404	T-Mobile Tuesdays: Win \$2,300 for 2023!	2022-12-27

	platform	library
175058	HTML5	arcade
176260	HTML5	arcade
175153	Flash	arcade
176537	HTML5	arcade
172001	HTML5	arcade
177431	Flash	theatre
176112	HTML5	arcade
175448	HTML5	arcade
172414	HTML5	arcade
172155	HTML5	arcade
176208	HTML5	arcade
176171	HTML5	arcade
176557	Lightning Strike	arcade
176901	HTML5	arcade
177091	HTML5	theatre
176747	HTML5	arcade
175851	HTML5	arcade
175933	HTML5	arcade
175934	HTML5	arcade
177404	HTML5	arcade

On the other side, here are the 20 most recent games. As expected, we find out that nowadays *HTML5* is the standard technology to make flash games, though some exceptions arise and there is even a *Flash* entry!

For the sake of completeness, let's restrict our search to *Flash*-only games.

```
[17]: df_dates.loc[(df_dates.platform == "Flash")][:20]
```

```
[17]:
```

	title	releaseDate	platform	\
175939	Claus.com	1995-01-01	Flash	
165097	Artcore	1996-01-01	Flash	
123757	The Simpsons 1996 website normal version	1996-01-01	Flash	
104886	The Hole in the Wall website intro	1996-01-01	Flash	
104785	Animated Screen Beans	1996-01-01	Flash	
105332	Past*Present*Future	1996-01-01	Flash	
97904	2 Design's Navigational Demo	1996-01-01	Flash	

98020	The Silicon Slip	1996-01-01	Flash
97944	Good Music Company Website	1996-01-01	Flash
99690	Simple, Tasty Buttons	1996-01-01	Flash
100122	Discrete Keyboard	1996-01-01	Flash
104809	HK Media Association website banner	1996-01-01	Flash
97943	FutureWave Software, Inc. Website	1996-01-01	Flash
97921	CHAOS Website	1996-01-01	Flash
98037	Zygomedia Website	1996-01-01	Flash
99628	First MouseOver Button	1996-01-01	Flash
106873	The Simpsons 1996 website	1996-01-01	Flash
104897	Volleynerd home page	1996-01-01	Flash
115310	Cloudberry Jam Puzzle	1997-01-01	Flash
167383	Disney 101 Dalmatians the Series microsite	1997-01-01	Flash

	library
175939	arcade
165097	arcade
123757	arcade
104886	theatre
104785	theatre
105332	theatre
97904	arcade
98020	arcade
97944	arcade
99690	arcade
100122	arcade
104809	theatre
97943	arcade
97921	arcade
98037	arcade
99628	arcade
106873	theatre
104897	theatre
115310	arcade
167383	arcade

The first *Flash* game is *Claus.com* from 1995. We notice that most of these are actually websites built in *Flash* and not exactly games or animations.

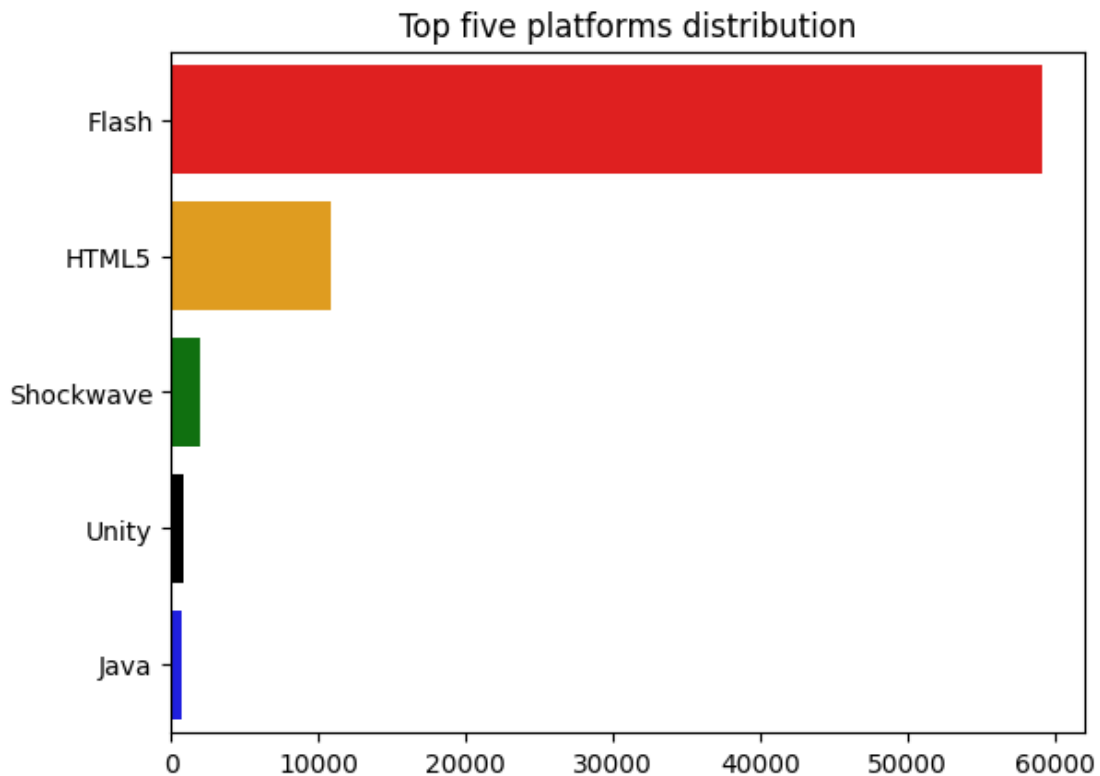
To take an overall view, let's compare the various platforms by games count, considering the top five.

```
[18]: top_platforms = df_dates["platform"].value_counts()[:5]
top_platforms
```

```
[18]: Flash          59047
HTML5             10904
Shockwave         1999
```

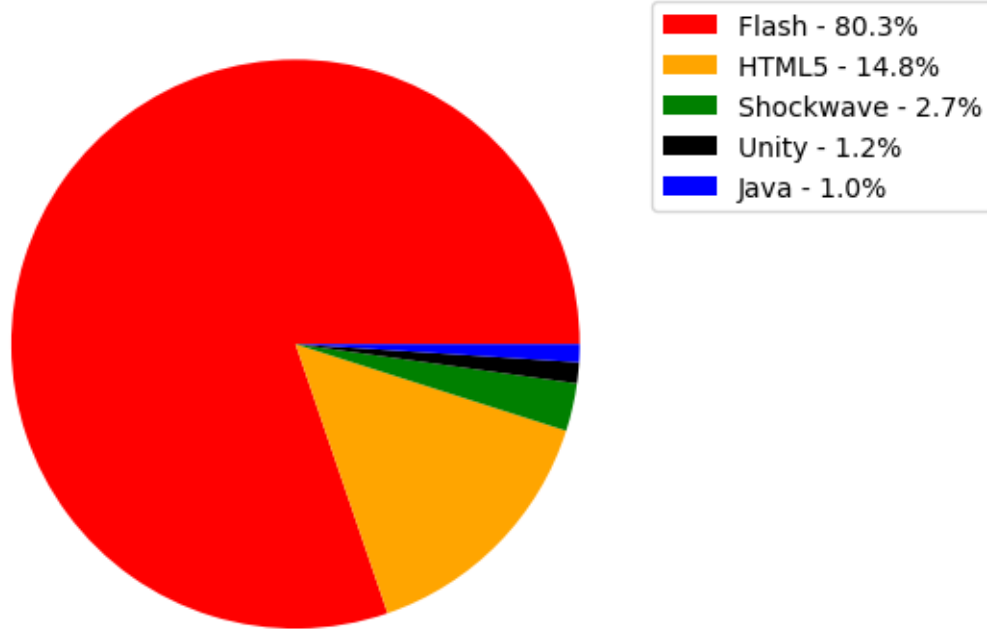
```
Unity          875
Java           751
Name: platform, dtype: int64
```

```
[19]: colors = ["red", "orange", "green", "black", "blue"]
sns.barplot(x = top_platforms.values, y = top_platforms.index, orient = "h",
            palette = colors).set(title = "Top five platforms distribution");
```



```
[20]: labels = top_platforms.index
sizes = top_platforms.values / top_platforms.values.sum() * 100
plt.pie(sizes, colors = colors, textprops = {"color": "w"})
labels = [f"{l} - {s:0.1f}%" for l, s in zip(labels, sizes)]
plt.legend(labels = labels, bbox_to_anchor = (1.5,1), loc = "best")
plt.title("Top five platforms distribution")
plt.show()
```

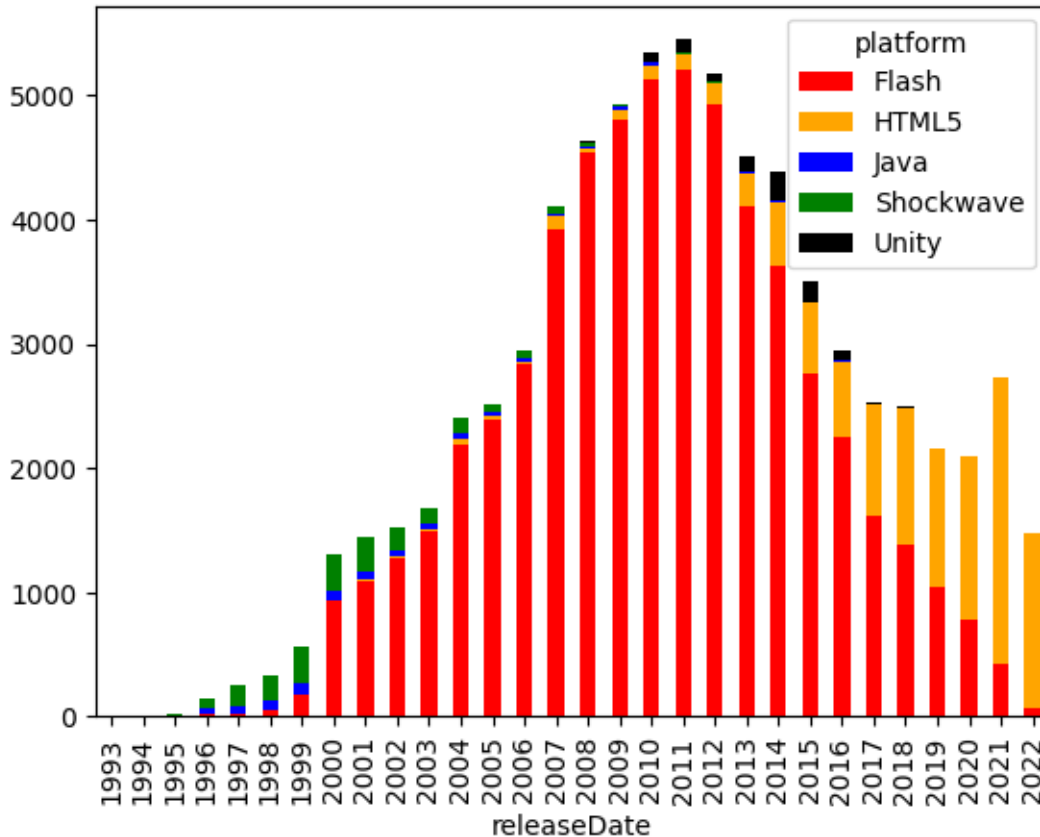
Top five platforms distribution



*Flash* is clearly the winner, followed by a rising *HTML5* and its old companion *Shockwave*, with *Unity* and *Java* as outsiders.

Web games were at their peak in the 2000s and many gamers are nostalgic about that decade, which could be considered a golden age. Thus, we expect to see that most of the games in our database have been released between 2000 and 2009. Let's check it out, while also comparing technologies against years.

```
[21]: df_year_platform = df_dates.loc[(df_dates["platform"].isin(top_platforms.
    ↪index)), :].copy()
years = df_year_platform["releaseDate"].astype(str).str[:4]
df_year_platform[df_year_platform.columns[1]] = years.values
df_year_platform.groupby(["releaseDate", "platform"]).size().unstack().
    ↪plot(kind = 'bar', stacked = True, color = ["red", "orange", "blue",
    ↪"green", "black"]);
```



*Et voila!* As we were expecting, web games have steadily risen in popularity in the first decade of the third millennium, reached a peak in 2011 and today their number is slowly decreasing, apart from some fluctuations. This does not mean that fewer games are being made: it could simply be that there is less incentive to curate and preserve a recent game written in a technology which will probably stay on for a long time compared to an old game with a nostalgic value that runs the risk to disappear at any time.

*Flash* dominated the scene between 2000 and 2017 (it's impressive to see that it lasted so long), while *HTML5* started to be relevant around 2013. *Shockwave* was most popular between 1996 and 2000, but continued to be used until 2007. *Unity* had six years of relative notoriety (2010-2016) and lastly, Java, despite being one of the first technologies eligible for making web games, has never known much use and moved off the radar around 2010.

## 1.5 Most common languages

Let's move on to another topic: *Flashpoint* allows non-English content as well, and it can be interesting to know which countries have contributed the most to the world of web games aside from the anglophone ones.

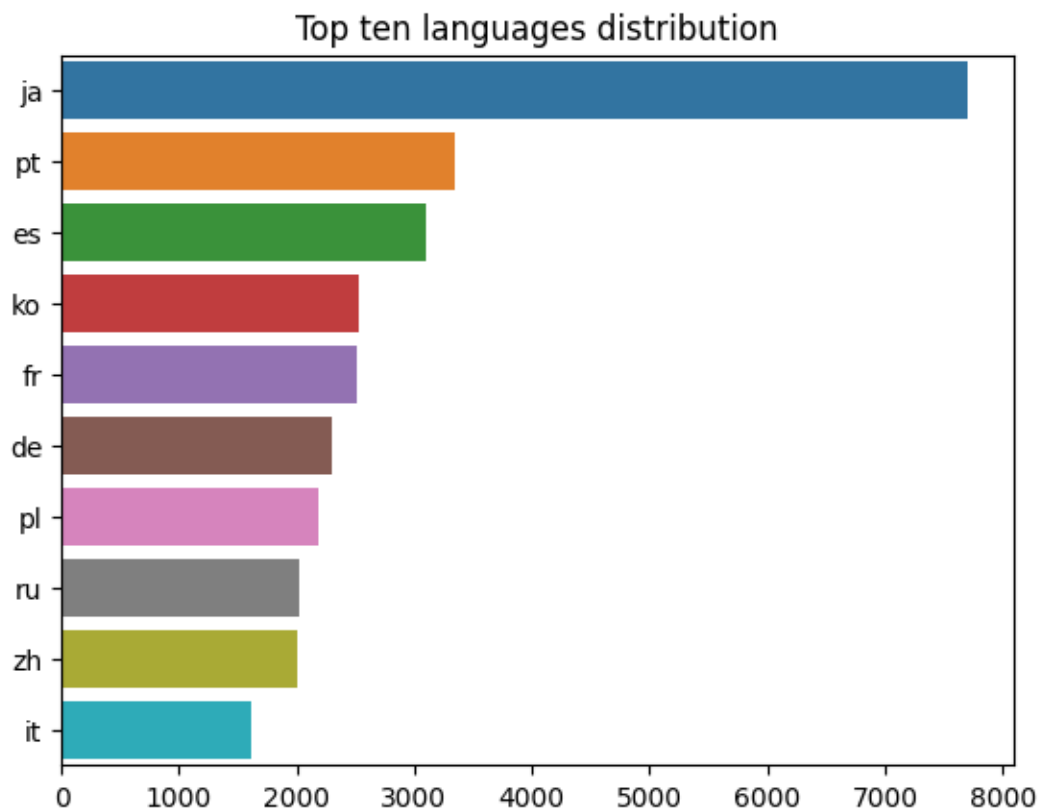
[22] :

```
top_languages = df.loc[df["language"] != "", "language"].str.replace(" ", ";").
    ↪str.split("; ").explode().value_counts().drop("en")[:10]
top_languages
```

```
[22]: ja    7709
      pt    3350
      es    3095
      ko    2525
      fr    2519
      de    2298
      pl    2183
      ru    2015
      zh    2011
      it    1612
      Name: language, dtype: int64
```

Since a game can come in different languages, separated by a colon and a space (“;”), we had to count each occurrence individually.

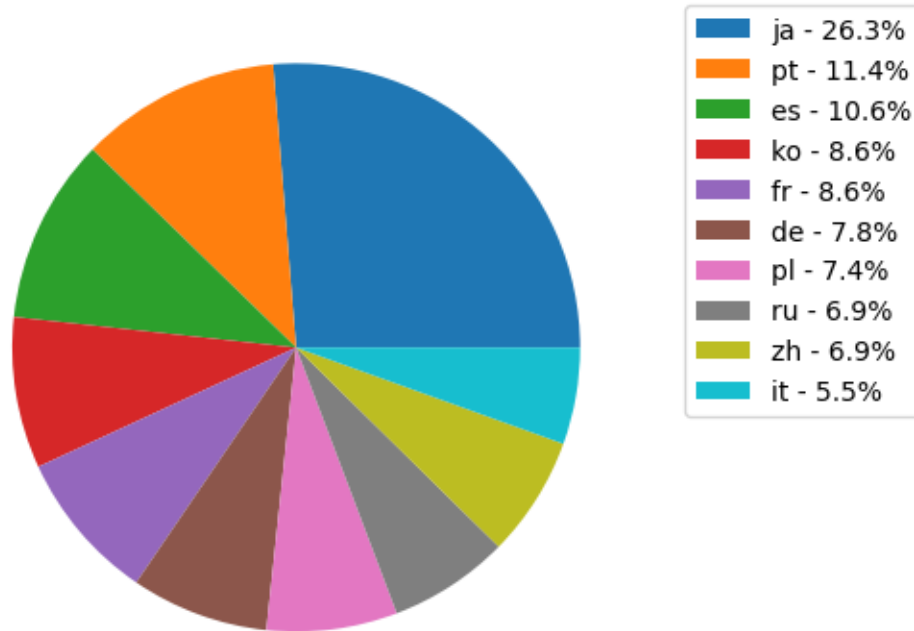
```
[23]: sns.barplot(x = top_languages.values, y = top_languages.index, orient = "h").
      ↪set(title = "Top ten languages distribution");
```





```
[24]: labels = top_languages.index
      sizes = top_languages.values / top_languages.values.sum() * 100
      plt.pie(sizes, textprops = {"color": "w"})
      labels = [f"{l} - {s:0.1f}%" for l, s in zip(labels, sizes)]
      plt.legend(labels = labels, bbox_to_anchor = (1.4,1), loc = "best")
      plt.title("Top ten languages distribution")
      plt.show()
```

Top ten languages distribution



We can see a strong presence of Asian content, with Japanese, Korean and Chinese among the top ten languages. The rest of the list is completed by European countries, namely Portugal, Spain, France, Germany, Poland and Italy, as well as Russia.

## 1.6 Most popular genres

Let's now focus on game genres, featured on the *tagsStr* column, to discover the most common ones.

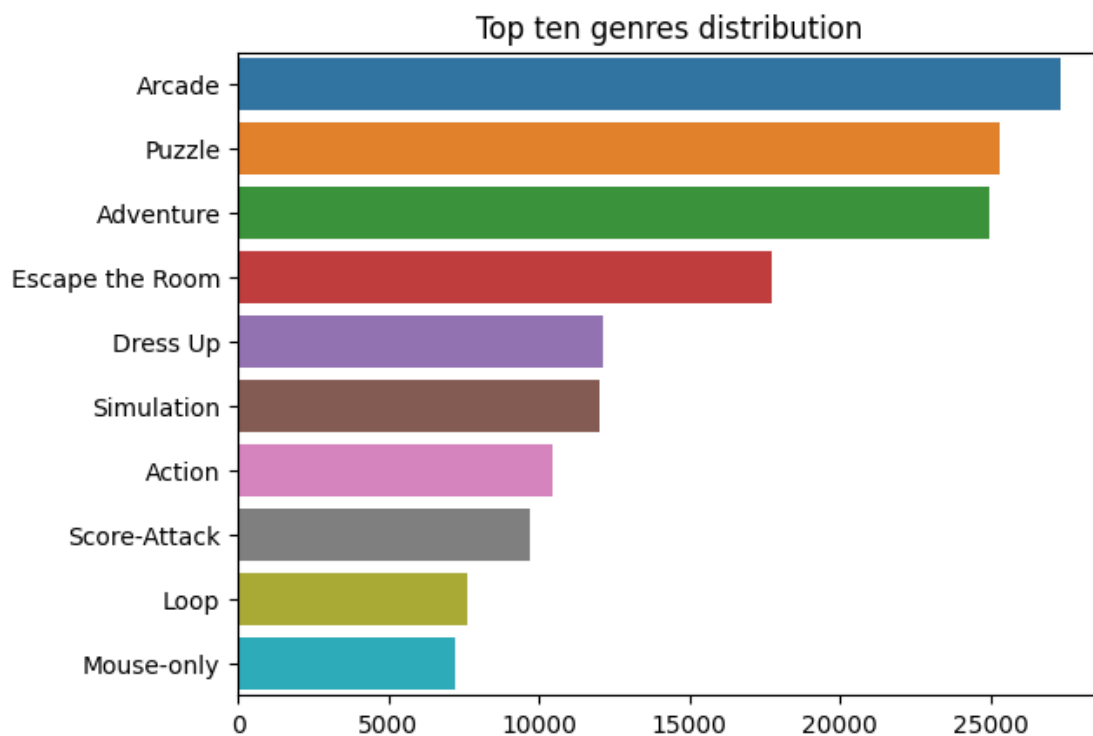
```
[25]: top_genres = df.loc[df["tagsStr"] != "", "tagsStr"].str.replace(",", ";").str.
      ↪split("; ").explode().value_counts()[:10]
      top_genres
```

```
[25]: Arcade          27308
      Puzzle          25284
```

Adventure	24981
Escape the Room	17725
Dress Up	12101
Simulation	11993
Action	10452
Score-Attack	9703
Loop	7599
Mouse-only	7232

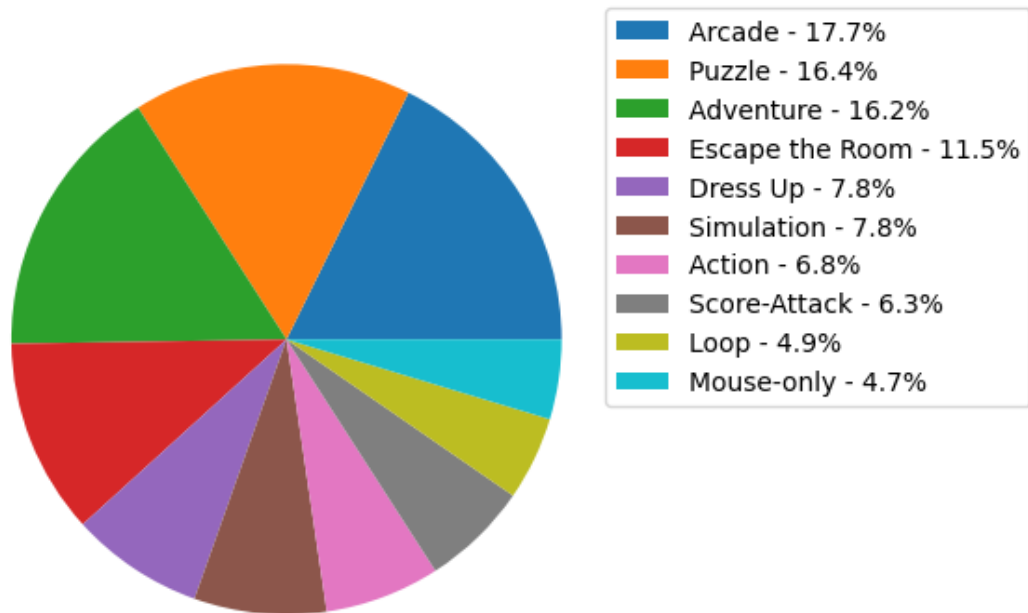
Name: tagsStr, dtype: int64

```
[26]: sns.barplot(x = top_genres.values, y = top_genres.index, orient = "h").
      ↪set(title = "Top ten genres distribution");
```



```
[27]: labels = top_genres.index
      sizes = top_genres.values / top_genres.values.sum() * 100
      plt.pie(sizes, textprops = {"color": "w"})
      labels = [f"{l} - {s:0.1f}%" for l, s in zip(labels, sizes)]
      plt.legend(labels = labels, bbox_to_anchor = (1.6, 1), loc = "best")
      plt.title("Top ten genres distribution")
      plt.show()
```

Top ten genres distribution



The big three genres are *Arcade*, *Puzzle* and *Adventure* and honestly it's kind of odd to see *Action* at such a low position. Conversely, as we expected from our previous analysis on developers, *Escape the Room* is fairly popular, along with *Dress Up* and *Simulation* games.

## 1.7 Most played games

As a final insight, let's find out which are the most played games among the *Flashpoint* users: to do this, we are going to use some official statistics from the platform itself. Visit <https://flashpoint-analytics.unstable.life/>, scroll down to the corresponding section and download the data in .csv format.

```
[28]: most_played = pd.read_csv("data/most_played.csv")
      most_played.rename(columns = {"category": "id"}, inplace = True)
      most_played
```

```
[28]:
```

	id	Play Count
0	83e1b5e7-4282-4bbd-868e-dcfa965e4abf	48493
1	a94d865c-cb38-4d31-96f3-dda26502c4a3	10228
2	617ca7f3-1cff-3f0c-5b53-07498b3b28d8	7454
3	8d09fc0d-6f25-4be6-b396-8fcaddad4e5e	7253
4	b0ce771e-7c02-4317-8528-ba48139e2688	6922
5	fdee4800-b5c9-49e0-b19e-22f2b0ccab68	6919
6	1e903a30-5c37-15bb-8e5e-6fea5a8103f2	6678

7	07921a2f-26fd-4364-9671-ee0c8d256ec1	6423
8	b9a8dbb9-0cd7-434b-b226-13dc9dd07b49	6341
9	a8707c0f-6aac-4c94-9d8a-cc397c97cc88	5745
10	92ba2d91-e041-2bd4-49ea-21758df711ff	5613
11	f55d6576-1aee-414a-bd1e-b7678d697dcf	5605
12	16b04977-f714-4239-b343-b759e16a33af	5579
13	5fa91cac-25c9-2a53-f391-09a099cb489b	5203
14	ed058412-fa96-4ea5-b2cb-2baff6a24b2a	5197
15	3337a4a2-dacf-4027-824c-aaf77623de65	4842
16	164e2dc8-31dd-4f59-a6be-46c2087e190f	4760
17	92781aea-6ad0-2bdb-a963-7131c49b7d07	4628
18	602762ba-a1b6-4810-8093-e5dbf0c33b61	4521
19	938f4383-8c92-8e8d-511b-004b5c69999d	4427
20	279b23ca-c7f3-4f56-a02f-3752ec7c5c5d	4407
21	6b50cb43-ad66-4ec8-9e46-3bb06821f2ab	4332
22	228087d3-c291-4a87-82e6-c6ffd3d4d2f6	4176
23	dbb21635-b0d5-78d9-a749-c4778a07e698	3958
24	de163cff-9dbe-fcbb-164c-53f5d6873fad	3883
25	16c27895-f7a6-6c65-18f3-feebcf87d28a	3874
26	c3ab2546-a7b0-89b2-82de-4044d61e1cbd	3684
27	b0d2b9a9-ab00-465e-b5a3-56031b92f070	3625
28	8f7f9fe8-4c55-43b1-a574-046c63712b39	3477
29	2ecf56d6-c5e4-a801-bc7f-60374ba1a051	3410
30	4423194a-22fe-427a-8eae-1b4d9c42395c	3247
31	9525910d-72b8-4e84-b668-43a267e00d9c	3238
32	c85ff4e3-1e2d-4b4e-8c4f-958d9db4aff2	3237
33	da6b3cb4-78a7-e998-fcf4-cb26a7950754	3227
34	2d0071b4-8ea9-40b1-a642-c970ca260cb2	3157
35	ab638461-9317-bbaa-9ccf-e7f0360e3b1b	3136
36	cf29b65d-e3e9-4da9-8201-d76de38736ea	3036
37	15ac0ed0-dbd4-8a95-5eec-4cf3e04fc771	3017
38	190c1bf6-4fcc-4b32-8278-f5b26db8eec1	2850
39	242dedc0-431a-4e4e-990c-58fe3c8ef740	2822

The file contains the *id* for the most 40 played games, along with a play count. Let's use the identifiers to find the titles of these games and their other info.

```
[29]: df_rank = df.merge(most_played, on = "id").sort_values("Play Count", ascending=
      ↪ False).reset_index()
df_rank
```

```
[29]:
```

	index	id \
0	39	83e1b5e7-4282-4bbd-868e-dcfa965e4abf
1	33	a94d865c-cb38-4d31-96f3-dda26502c4a3
2	26	617ca7f3-1cff-3f0c-5b53-07498b3b28d8
3	18	8d09fc0d-6f25-4be6-b396-8fcaddad4e5e
4	1	b0ce771e-7c02-4317-8528-ba48139e2688
5	35	fdee4800-b5c9-49e0-b19e-22f2b0ccab68

6	3	1e903a30-5c37-15bb-8e5e-6fea5a8103f2
7	22	07921a2f-26fd-4364-9671-ee0c8d256ec1
8	30	b9a8dbb9-0cd7-434b-b226-13dc9dd07b49
9	9	a8707c0f-6aac-4c94-9d8a-cc397c97cc88
10	4	92ba2d91-e041-2bd4-49ea-21758df711ff
11	24	f55d6576-1aee-414a-bd1e-b7678d697dcf
12	19	16b04977-f714-4239-b343-b759e16a33af
13	0	5fa91cac-25c9-2a53-f391-09a099cb489b
14	37	ed058412-fa96-4ea5-b2cb-2baff6a24b2a
15	36	3337a4a2-dacf-4027-824c-aaf77623de65
16	8	164e2dc8-31dd-4f59-a6be-46c2087e190f
17	11	92781aea-6ad0-2bdb-a963-7131c49b7d07
18	20	602762ba-a1b6-4810-8093-e5dbf0c33b61
19	27	938f4383-8c92-8e8d-511b-004b5c69999d
20	31	279b23ca-c7f3-4f56-a02f-3752ec7c5c5d
21	2	6b50cb43-ad66-4ec8-9e46-3bb06821f2ab
22	15	228087d3-c291-4a87-82e6-c6ffd3d4d2f6
23	38	dbb21635-b0d5-78d9-a749-c4778a07e698
24	17	de163cff-9dbe-fcbb-164c-53f5d6873fad
25	16	16c27895-f7a6-6c65-18f3-feebcf87d28a
26	23	c3ab2546-a7b0-89b2-82de-4044d61e1cbd
27	5	b0d2b9a9-ab00-465e-b5a3-56031b92f070
28	32	8f7f9fe8-4c55-43b1-a574-046c63712b39
29	10	2ecf56d6-c5e4-a801-bc7f-60374ba1a051
30	13	4423194a-22fe-427a-8eae-1b4d9c42395c
31	29	9525910d-72b8-4e84-b668-43a267e00d9c
32	12	c85ff4e3-1e2d-4b4e-8c4f-958d9db4aff2
33	25	da6b3cb4-78a7-e998-fcf4-cb26a7950754
34	34	2d0071b4-8ea9-40b1-a642-c970ca260cb2
35	21	ab638461-9317-bbaa-9ccf-e7f0360e3b1b
36	6	cf29b65d-e3e9-4da9-8201-d76de38736ea
37	14	15ac0ed0-dbd4-8a95-5eec-4cf3e04fc771
38	7	190c1bf6-4fcc-4b32-8278-f5b26db8eec1
39	28	242dedc0-431a-4e4e-990c-58fe3c8ef740

	title \
0	Poptropica
1	Jacksmith
2	Papa's Cheeseria
3	Strike Force Heroes 3
4	Super Mario Bros. Crossover
5	Strike Force Heroes
6	Papa's Scooperia
7	Bloons TD 5
8	Papa's Sushiria
9	Papa's Bakeria
10	Papa's Freezeria

11	Papa's Burgeria
12	Super Mario 63
13	Epic Battle Fantasy 5
14	Super Smash Flash 2
15	Flappy Bird For Dinner
16	Madness: Project Nexus
17	Papa's Hot Doggeria
18	Swords and Souls
19	Papa's Donuteria
20	Papa's Pizzeria
21	Strike Force Heroes 2
22	The Impossible Quiz
23	Ben 10: Battle Ready
24	Papa's Cupcakeria
25	Papa Louie 2: When Burgers Attack!
26	Papa's Wingeria
27	Portal: The Flash Version
28	The Last Stand: Union City
29	Papa's Pastaria
30	Cactus McCoy
31	Papa's Pancakeria
32	Age of War
33	Plants vs Zombies (Web Version)
34	Bowman
35	Papa Louie 3: When Sundaes Attack!
36	Electricman 2 - The Tournament of Voltagen
37	Swords and Sandals 2
38	Road of the Dead
39	Commando 2

	developer	publisher	platform \
0	Sandbox Networks Inc.; Pearson Education		Flash
1	Flipline Studios	PapaLouie.com	Flash
2	Flipline Studios	PapaLouie.com	Flash
3	Sky9 Games	Armor Games	Flash
4	Exploding Rabbit		Flash
5	Sky9 Games	Armor Games	Flash
6	Flipline Studios	PapaLouie.com	Flash
7	Ninja Kiwi	Ninja Kiwi	Flash
8	Flipline Studios	PapaLouie.com	Flash
9	Flipline Studios	PapaLouie.com	Flash
10	Flipline Studios	Armor Games	Flash
11	Flipline Studios		Flash
12	Runouw	Newgrounds	Flash
13	Kupo Games		Flash
14	McLeodGaming	McLeodGaming	Flash
15		Cooking Games	Flash

16	Michael Swain & Matt Krinkels Jolly	Newgrounds	Flash
17	Flipline Studios	PapaLouie.com	Flash
18	SoulGame	Armor Games	Flash
19	Flipline Studios	PapaLouie.com	Flash
20	Flipline Studios		Flash
21	Sky9 Games	Armor Games	Flash
22	Splapp-Me-Do	Newgrounds	Flash
23		Cartoon Network	Shockwave
24	Flipline Studios	PapaLouie.com	Flash
25	Flipline Studios	PapaLouie.com	Flash
26	Flipline Studios	PapaLouie.com	Flash
27	We Create Stuff	Armor Games	Flash
28	ConArtists	Armor Games	Flash
29	Flipline Studios	PapaLouie.com	Flash
30	Flipline Studios		Flash
31	Flipline Studios	PapaLouie.com	Flash
32	Louissi	Max Games	Flash
33	PopCap		Flash
34	flash		Flash
35	Flipline Studios	PapaLouie.com	Flash
36	Damien Clarke	Armor Games	Flash
37	3rdsense.com		Flash
38	Evil Dog	Newgrounds	Flash
39	Miniclip.com	Miniclip.com	Flash

	releaseDate	language	library	\
0	2007-09	en	arcade	
1	2012-09-27	en	arcade	
2	2015-06-10	en	arcade	
3		en	arcade	
4	2010-04-27	en	arcade	
5	2012-05-31	en	arcade	
6	2018-07-24	en	arcade	
7	2011-12-13	en	arcade	
8	2016-12-13	en	arcade	
9	2016-03-14	en	arcade	
10	2011-08-05	en	arcade	
11	2010-12-06	en	arcade	
12	2009-06-10	en	arcade	
13	2020-01-21		arcade	
14	2007-12-25	en	arcade	
15		en	arcade	
16	2012-03-25	en	arcade	
17	2012-11-19	en	arcade	
18	2015-10-18	en	arcade	
19	2014-06-16	en	arcade	
20	2007-08-07	en	arcade	

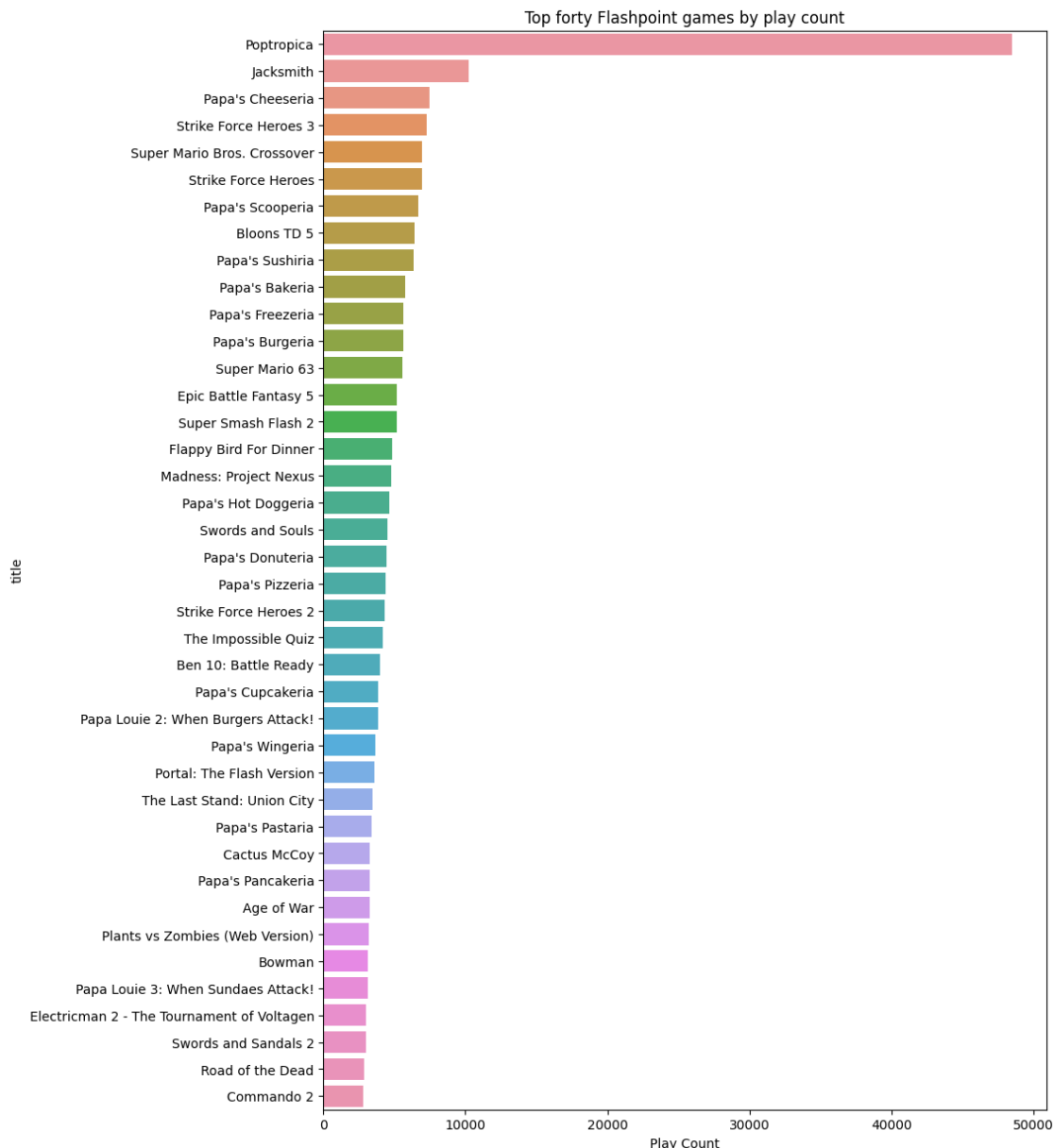
21		en	arcade
22	2007-02-20	en	arcade
23		en; ja	arcade
24	2013-08-07	en	arcade
25	2013-06-05	en	arcade
26	2012-06-13	en	arcade
27	2007-11-19	en	arcade
28	2011-07-29		arcade
29	2013-12-09	en	arcade
30	2011-03-10	en	arcade
31	2012-03-05	en	arcade
32	2008-01-14	en	arcade
33		en	arcade
34	2004-12-05	en	arcade
35	2015-03-04	en	arcade
36	2008-01-08	en	arcade
37	2007-01-07	en	arcade
38	2010-10-13	en	arcade
39	2008-10-03		arcade

	tagsStr	Play Count
0	Educational; Platformer; Puzzle; Variety; Side...	48493
1	Simulation; Time Management	10228
2	Cooking; Simulation; Time Management; Achievem...	7454
3	Action; Shooter	7253
4	Arcade; Platformer; Score-Attack; Pixel; Side-...	6922
5	Shooter; Action	6919
6	Cooking; Simulation; Time Management; Achievem...	6678
7	Tower Defense; Strategy	6423
8	Cooking; Simulation; Time Management; Achievem...	6341
9	Cooking; Simulation; Time Management; Achievem...	5745
10	Cooking; Simulation; Time Management; Achievem...	5613
11	Cooking; Simulation; Time Management	5605
12	Level Editor; Platformer; Super Mario	5579
13	RPG	5203
14	Bleach; Bomberman; Chibi-Robo!; F-Zero; Fire E...	5197
15	Cooking; Simulation	4842
16	Action; Beat 'Em Up; Shooter; 3/4 View; Madnes...	4760
17	Cooking; Simulation; Time Management; Achievem...	4628
18	Role-Playing	4521
19	Cooking; Simulation; Time Management; Achievem...	4427
20	Cooking; Simulation; Time Management	4407
21	Shooter; Action	4332
22	Quiz	4176
23	Action; Ben 10	3958
24	Cooking; Simulation; Time Management; Achievem...	3883
25	Platformer; Side-Scrolling	3874



26	Cooking; Simulation; Time Management; Achievem...	3684
27	Platformer; Puzzle; Portal	3625
28	Shooter	3477
29	Cooking; Simulation; Time Management; Achievem...	3410
30	Platformer	3247
31	Cooking; Simulation; Time Management; Achievem...	3238
32	Strategy; Mouse-only	3237
33	Strategy; Tower Defense; 3/4 View; Plants vs. ...	3227
34	Action; Archery; Artillery; Blood	3157
35	Platformer; Side-Scrolling	3136
36	Action; Beat 'Em Up; Stick	3036
37	Role-Playing	3017
38	Action; Driving; Zombie; First-Person; Upgrade	2850
39	Shooter	2822

```
[30]: fig, ax = plt.subplots(figsize=(10, 15))
sns.barplot(x = df_rank["Play Count"][:40], y = df_rank["title"][:40], orient = "h", ax = ax).set(title = "Top forty Flashpoint games by play count");
```



There we go! *Poptropica* is the indisputable winner, with almost fifty thousand play counts. There is a massive presence of *Papa's Gamera* franchise, as well as all-time classics like *Strike Force Heroes*, *Super Mario 63* and *Age of War*. Finally, a special remark about *Ben 10: Battle Ready*, which was thought to be lost forever, before it was restored and made playable again on *Flashpoint*.

## 1.8 Conclusion

This was a thorough analysis of the *Flashpoint* catalogue, which hopefully gives some insights about the world of web-based games and their significant relevance in the history of the Internet.

The effort to preserve this kind of content has generated amazing results, saving an astounding

quantity of material which would have disappeared otherwise. Despite the concrete risk of a digital dark age, we should insist on preserving the stuff that we care about and keep it alive, not only for historical reasons, but also for the nostalgic value we associate with it.