Задача:

Сформируйте модель монетизации игрового приложения.

Многие игры зарабатывают с помощью рекламы. И все они сталкиваются с противоречием:

- Пользователь разозлится и уйдёт, если начать показывать ему рекламу раньше, чем игра его затянет.
- Но чем позже создатели игры включат рекламу, тем меньше они заработают.

Аналитик помогает бизнесу выбрать оптимальное время для запуска рекламы. Зная расходы на продвижение игры, он может рассчитать её окупаемость при разных сценариях

Пока создатели игры планируют показывать её на экране выбором постройки. Помогите им не уйти в минус.

1. Предобработка данных

1. Исследовательский анализ данных

- количество игроков, перешедших на 1 уровень победив врага
- метрики монетизации:
  - DAU, WAU
- график по событиям, включая игроков перешедших на 1 уровень победив врага
- график по количеству объектов
- график по реализованным проектам
- построить график по дням, по которому произошел клик по объявлению
- график для источников, с которых пришел пользователь

1. Статистические гипотезы

- Проверьте гипотезу различия времени прохождения уровня между пользователями, которые заканчивают уровень через реализацию проекта, и пользователями, которые заканчивают уровень победой над другим игроком.

Сформулируйте и проверьте статистическую гипотезу относительно представленных данных:

- Проверить различие кто больше приносит денег по кликам - пользователи, которые заканчивают уровень "побеждая врага" или пользователи, которые заканчивают уровень через реализацию проекта

1. Выводы

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt

import seaborn as sns
sns.set(rc={'figure.figsize':(10, 8)})

import scipy.stats as stats
from scipy import stats as st

import math as mth

import numpy as np

import pandas as pdm
from datetime import datetime,timedelta

from pathlib import Path
import matplotlib.dates as mdates

import math
import cmath
```

In [2]:
```
ad_costs = pd.read_csv('/datasets/ad_costs.csv', sep=',')
user_source = pd.read_csv('/datasets/user_source.csv', sep=',')
game_actions = pd.read_csv('/datasets/game_actions.csv', sep=',')
```

In [3]:
```
ad_costs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28 entries, 0 to 27
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   source  28 non-null     object
```

```
 1   day     28 non-null     object
 2   cost    28 non-null     float64
dtypes: float64(1), object(2)
memory usage: 800.0+ bytes
```

In [4]:
```
user_source.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13576 entries, 0 to 13575
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   user_id  13576 non-null  object
 1   source   13576 non-null  object
dtypes: object(2)
memory usage: 212.2+ KB
```

In [5]:
```
game_actions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 135640 entries, 0 to 135639
Data columns (total 5 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   event_datetime  135640 non-null  object
 1   event           135640 non-null  object
 2   building_type   127957 non-null  object
 3   user_id         135640 non-null  object
 4   project_type    1866 non-null    object
dtypes: object(5)
memory usage: 5.2+ MB
```

In [6]:
```
ad_costs.duplicated().sum()
```

Out[6]:
```
0
```

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

Out[7]:
```
1
```

In [8]:
```
game_actions = game_actions.drop_duplicates()
```

In [9]:
```
user_source.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
ad_costs['day'] = pd.to_datetime(ad_costs['day'])
```

In [11]:
```
game_actions.head()
```

Out[11]:

	event_datetime	event	building_type	user_id	project_type
0	2020-05-04 00:00:01	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	NaN
1	2020-05-04 00:00:03	building	assembly_shop	c07b1c10-f477-44dc-81dc-ec82254b1347	NaN
2	2020-05-04 00:00:16	building	assembly_shop	6edd42cc-e753-4ff6-a947-2107cd560710	NaN
3	2020-05-04 00:00:16	building	assembly_shop	92c69003-d60a-444a-827f-8cc51bf6bf4c	NaN
4	2020-05-04 00:00:35	building	assembly_shop	cdc6bb92-0ccb-4490-9866-ef142f09139d	NaN

In [12]:
```
game_actions['project_type'].unique()
```

Out[12]:
```
array([nan, 'satellite_orbital_assembly'], dtype=object)
```

In [13]:
```
game_actions['building_type'].unique()
```

Out[13]:
```
array(['assembly_shop', 'spaceport', nan, 'research_center'], dtype=object)
```

могу предположить что пропуски в building_type являются следствием того что здания просто напросто непостроены, а project_type следствие того что орбитальная станция не построена

In [14]:
```
game_actions['project_type'] = game_actions['project_type'].fillna('unknown')
```

In [15]:
```
game_actions['building_type'] = game_actions['building_type'].fillna('unknown')
```

In [16]:
```
game_actions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 135639 entries, 0 to 135639
Data columns (total 5 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   event_datetime  135639 non-null  object
 1   event           135639 non-null  object
 2   building_type   135639 non-null  object
 3   user_id         135639 non-null  object
 4   project_type    135639 non-null  object
dtypes: object(5)
memory usage: 6.2+ MB
```

In [17]:
```
game_actions['time'] = pd.to_datetime(game_actions['event_datetime'])
game_actions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 135639 entries, 0 to 135639
Data columns (total 6 columns):
 #   Column          Non-Null Count   Dtype         
---  ------          --------------   -----         
 0   event_datetime  135639 non-null  object        
 1   event           135639 non-null  object        
 2   building_type   135639 non-null  object        
 3   user_id         135639 non-null  object        
 4   project_type    135639 non-null  object        
 5   time            135639 non-null  datetime64[ns]
dtypes: datetime64[ns](1), object(5)
memory usage: 7.2+ MB
```

In [18]:
```
game_actions.sample(5)
```

Out[18]:

	event_datetime	event	building_type	user_id	project_type	time
39688	2020-05-08 03:24:59	building	spaceport	a638b258-c2b8-4f66-bd17-5afc6ab06ba3	unknown	2020-05-08 03:24:59
78321	2020-05-10 23:52:24	building	spaceport	b1a84002-f99b-4cc4-b490-039d3aea42c6	unknown	2020-05-10 23:52:24
72679	2020-05-10 15:23:27	finished_stage_1	unknown	f63bb9b9-ddea-43d3-bad6-c0c879ef008f	unknown	2020-05-10 15:23:27

	event_datetime	event	building_type	user_id	project_type	time
72711	2020-05-10 15:26:13	building	assembly_shop	df18d3a3-bec4-4f31-9a11-07ea6d8f28e9	unknown	2020-05-10 15:26:13
73098	2020-05-10 16:01:52	building	assembly_shop	1a1c8bc0-589e-4e88-bc13-cf8717603d43	unknown	2020-05-10 16:01:52

In [19]:
```
game_actions['project_type'].unique()
```

Out[19]:
```
array(['unknown', 'satellite_orbital_assembly'], dtype=object)
```

In [20]:
```
game_actions['building_type'].unique()
```

Out[20]:
```
array(['assembly_shop', 'spaceport', 'unknown', 'research_center'],
      dtype=object)
```

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

Out[21]:

	user_id	source
0	0001f83c-c6ac-4621-b7f0-8a28b283ac30	facebook_ads
1	00151b4f-ba38-44a8-a650-d7cf130a0105	yandex_direct
2	001aaea6-3d14-43f1-8ca8-7f48820f17aa	youtube_channel_reklama
3	001d39dc-366c-4021-9604-6a3b9ff01e25	instagram_new_adverts
4	002f508f-67b6-479f-814b-b05f00d4e995	facebook_ads

In [22]:
```
ad_costs.sample(5)
```

Out[22]:

	source	day	cost
27	youtube_channel_reklama	2020-05-09	23.314669
25	youtube_channel_reklama	2020-05-07	55.740645
17	yandex_direct	2020-05-06	180.917099
13	instagram_new_adverts	2020-05-09	46.775400

	source	day	cost
9	instagram_new_adverts	2020-05-05	313.970984

In [23]:
```
ad_costs.describe()
```

Out[23]:

	cost
count	28.000000
mean	271.556321
std	286.867650
min	23.314669
25%	66.747365
50%	160.056443
75%	349.034473
max	969.139394

ничего больше интересного в предобработке нет

Исследовательский анализ данных

количество игроков, перешедших на 1 уровень победив врага

In [24]:
```
game_actions['event'].unique()
```

Out[24]:
```
array(['building', 'finished_stage_1', 'project'], dtype=object)
```

In [25]:
```
project_finished = game_actions.query('event == ("project", "finished_stage_1")')
```

In [26]:
```
project_finished['count'] = project_finished['user_id'].map(project_finished['user_id'].value_counts())
#project_finished['count'] = project_finished.groupby('user_id')['event'].count().reset_index().transform('count')
project_finished
```

```
/tmp/ipykernel_2375/1049838982.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  project_finished['count'] = project_finished['user_id'].map(project_finished['user_id'].value_counts())
```

Out[26]:

	event_datetime	event	building_type	user_id	project_type	time	count
6659	2020-05-04 19:47:29	finished_stage_1	unknown	ced7b368-818f-48f6-9461-2346de0892c5	unknown	2020-05-04 19:47:29	1
13134	2020-05-05 13:22:09	finished_stage_1	unknown	7ef7fc89-2779-46ea-b328-9e5035b83af5	unknown	2020-05-05 13:22:09	1
15274	2020-05-05 18:54:37	finished_stage_1	unknown	70db22b3-c2f4-43bc-94ea-51c8d2904a29	unknown	2020-05-05 18:54:37	1
16284	2020-05-05 21:27:29	finished_stage_1	unknown	903fc9ef-ba97-4b12-9d5c-ac8d602fbd8b	unknown	2020-05-05 21:27:29	1
19650	2020-05-06 06:02:22	finished_stage_1	unknown	58e077ba-feb1-4556-a5a0-d96bd04efa39	unknown	2020-05-06 06:02:22	1
...	...	...	...	...	...	...	...
135632	2020-06-04 15:50:38	finished_stage_1	unknown	22cce310-fe10-41a2-941b-9c3d63327fea	unknown	2020-06-04 15:50:38	1
135633	2020-06-04 17:56:14	finished_stage_1	unknown	d477dde8-7c22-4f23-9c4f-4ec31a1aa4c8	unknown	2020-06-04 17:56:14	2
135636	2020-06-05 02:25:12	finished_stage_1	unknown	515c1952-99aa-4bca-a7ea-d0449eb5385a	unknown	2020-06-05 02:25:12	1
135638	2020-06-05 12:12:27	finished_stage_1	unknown	32572adb-900f-4b5d-a453-1eb1e6d88d8b	unknown	2020-06-05 12:12:27	1
135639	2020-06-05 12:32:49	finished_stage_1	unknown	f21d179f-1c4b-437e-b9c6-ab1976907195	unknown	2020-06-05 12:32:49	1

7683 rows × 7 columns

In [27]:
```
project_finished['count'] = project_finished['count'].astype(int)
```

```
/tmp/ipykernel_2375/1310609038.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  project_finished['count'] = project_finished['count'].astype(int)
```

In [28]:
```
project_finished['count'].unique()
```

Out[28]:
```
array([1, 2])
```

In [29]:
```
project_finished['count'] = ['warrior' if x == 1 else 'builder' for x in project_finished['count']]
```

```
/tmp/ipykernel_2375/3843735455.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  project_finished['count'] = ['warrior' if x == 1 else 'builder' for x in project_finished['count']]
```

In [30]:
```
project_finished['count'].unique()
```

Out[30]:
```
array(['warrior', 'builder'], dtype=object)
```

In [31]:
```
project_finished.head()
```

Out[31]:

	event_datetime	event	building_type	user_id	project_type	time	count
6659	2020-05-04 19:47:29	finished_stage_1	unknown	ced7b368-818f-48f6-9461-2346de0892c5	unknown	2020-05-04 19:47:29	warrior
13134	2020-05-05 13:22:09	finished_stage_1	unknown	7ef7fc89-2779-46ea-b328-9e5035b83af5	unknown	2020-05-05 13:22:09	warrior
15274	2020-05-05 18:54:37	finished_stage_1	unknown	70db22b3-c2f4-43bc-94ea-51c8d2904a29	unknown	2020-05-05 18:54:37	warrior
16284	2020-05-05 21:27:29	finished_stage_1	unknown	903fc9ef-ba97-4b12-9d5c-ac8d602fbd8b	unknown	2020-05-05 21:27:29	warrior
19650	2020-05-06 06:02:22	finished_stage_1	unknown	58e077ba-feb1-4556-a5a0-d96bd04efa39	unknown	2020-05-06 06:02:22	warrior

In [32]:
```
df1 = pd.merge(game_actions, project_finished, how = 'left')
df1 = df1.dropna()
df1
```

Out[32]:

	event_datetime	event	building_type	user_id	project_type	time	count
6659	2020-05-04 19:47:29	finished_stage_1	unknown	ced7b368-818f-48f6-9461-2346de0892c5	unknown	2020-05-04 19:47:29	warrior

	event_datetime	event	building_type	user_id	project_type	time	count
13134	2020-05-05 13:22:09	finished_stage_1	unknown	7ef7fc89-2779-46ea-b328-9e5035b83af5	unknown	2020-05-05 13:22:09	warrior
15274	2020-05-05 18:54:37	finished_stage_1	unknown	70db22b3-c2f4-43bc-94ea-51c8d2904a29	unknown	2020-05-05 18:54:37	warrior
16284	2020-05-05 21:27:29	finished_stage_1	unknown	903fc9ef-ba97-4b12-9d5c-ac8d602fbd8b	unknown	2020-05-05 21:27:29	warrior
19650	2020-05-06 06:02:22	finished_stage_1	unknown	58e077ba-feb1-4556-a5a0-d96bd04efa39	unknown	2020-05-06 06:02:22	warrior
...	...	...	...	...	...	...	...
135631	2020-06-04 15:50:38	finished_stage_1	unknown	22cce310-fe10-41a2-941b-9c3d63327fea	unknown	2020-06-04 15:50:38	warrior
135632	2020-06-04 17:56:14	finished_stage_1	unknown	d477dde8-7c22-4f23-9c4f-4ec31a1aa4c8	unknown	2020-06-04 17:56:14	builder
135635	2020-06-05 02:25:12	finished_stage_1	unknown	515c1952-99aa-4bca-a7ea-d0449eb5385a	unknown	2020-06-05 02:25:12	warrior
135637	2020-06-05 12:12:27	finished_stage_1	unknown	32572adb-900f-4b5d-a453-1eb1e6d88d8b	unknown	2020-06-05 12:12:27	warrior
135638	2020-06-05 12:32:49	finished_stage_1	unknown	f21d179f-1c4b-437e-b9c6-ab1976907195	unknown	2020-06-05 12:32:49	warrior

7683 rows × 7 columns

метрики монетизации:

- DAU, WAU

In [33]:
```
game_actions = game_actions.rename(columns={'event_datetime': 'day'})
```

In [34]:
```
df = pd.merge(game_actions, user_source, how = 'left')

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 135639 entries, 0 to 135638
Data columns (total 7 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   day            135639 non-null  object
 1   event          135639 non-null  object
 2   building_type  135639 non-null  object
 3   user_id        135639 non-null  object
```

```
 4   project_type   135639 non-null  object
 5   time           135639 non-null  datetime64[ns]
 6   source         135639 non-null  object
dtypes: datetime64[ns](1), object(6)
memory usage: 8.3+ MB
```

In [35]:
```
df.head()
```

Out[35]:

	day	event	building_type	user_id	project_type	time	source
0	2020-05-04 00:00:01	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	unknown	2020-05-04 00:00:01	youtube_channel_reklama
1	2020-05-04 00:00:03	building	assembly_shop	c07b1c10-f477-44dc-81dc-ec82254b1347	unknown	2020-05-04 00:00:03	facebook_ads
2	2020-05-04 00:00:16	building	assembly_shop	6edd42cc-e753-4ff6-a947-2107cd560710	unknown	2020-05-04 00:00:16	instagram_new_adverts
3	2020-05-04 00:00:16	building	assembly_shop	92c69003-d60a-444a-827f-8cc51bf6bf4c	unknown	2020-05-04 00:00:16	facebook_ads
4	2020-05-04 00:00:35	building	assembly_shop	cdc6bb92-0ccb-4490-9866-ef142f09139d	unknown	2020-05-04 00:00:35	yandex_direct

In [36]:
```
df['day'] = df['day'].astype('datetime64')
```

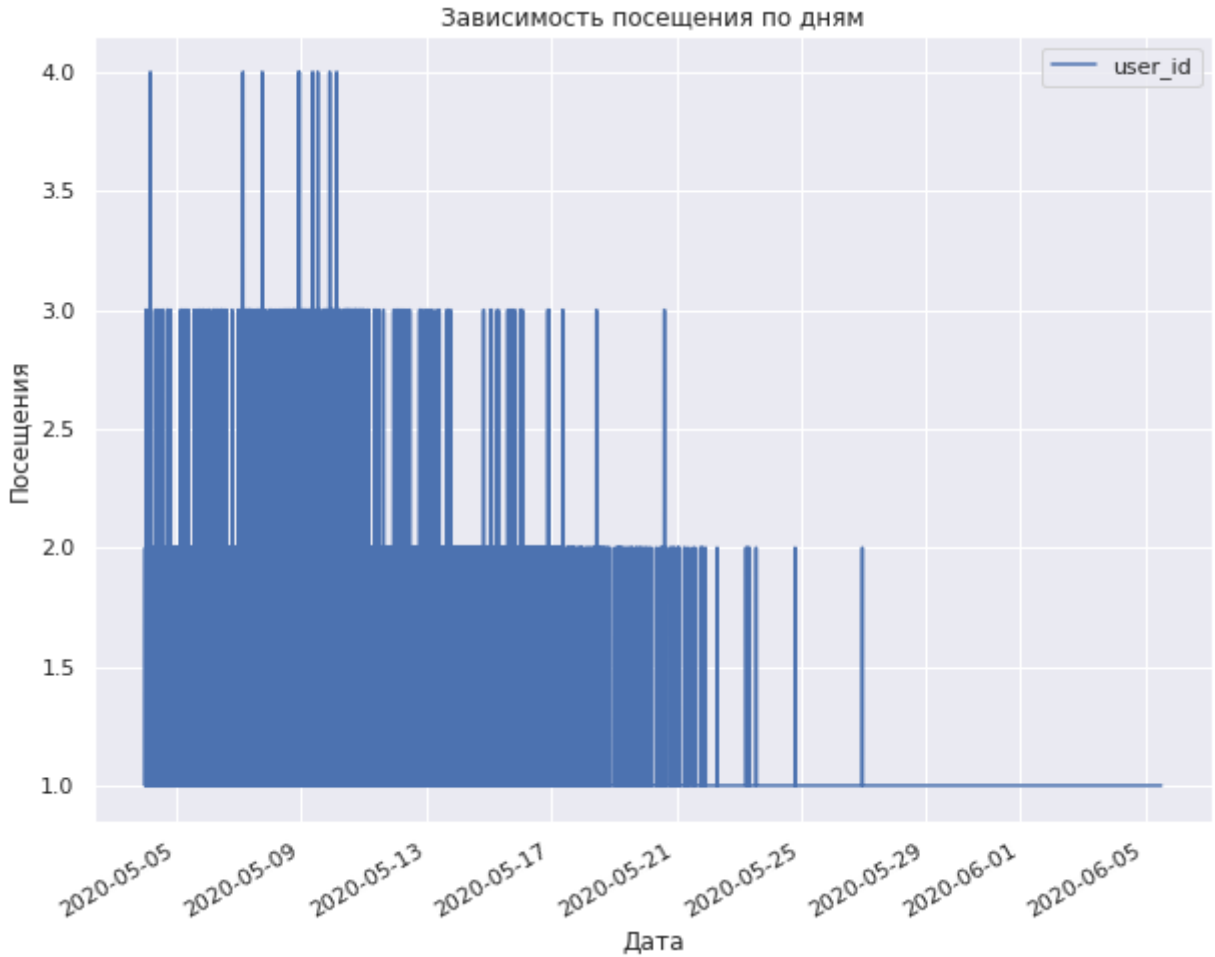
In [37]:
```
df['week'] = df['day'].dt.week
```

```
/tmp/ipykernel_2375/2457541713.py:1: FutureWarning: Series.dt.weekofyear and Series.dt.week have been deprecated.  Please use Seri
es.dt.isocalendar().week instead.
  df['week'] = df['day'].dt.week
```

In [38]:
```
dau = df.groupby('day').agg({'user_id': 'nunique'})
```

In [39]:
```
ax_dau = dau.plot()
ax_dau.set_title('Зависимость посещения по дням')
ax_dau.set_xlabel('Дата')
ax_dau.set_ylabel('Посещения')
```

Out[39]:
```
Text(0, 0.5, 'Посещения')
```



видим что все меньше пользователей получает новый уровень

In [40]:
```
wau = df.groupby('week').agg({'user_id': 'nunique'})
wau
```

Out[40]:

week	user_id
19	13576
20	12121

week	user_id
21	4353
22	521
23	29

график по событиям, включая игроков перешедших на 1 уровень победив врага

In [41]:
```
import plotly.graph_objects as go
```

In [42]:
```
pie = df.groupby('event')['user_id'].count().reset_index()

labels = pie.event
values = pie.user_id

fig = go.Figure(data=[go.Pie(labels=labels, values=values,hole=0.3)])
fig.update_traces(hoverinfo='label+percent', textinfo='label+value+percent'
                  )
fig.update_layout(
    title_text="Проекты")
fig.show()
```

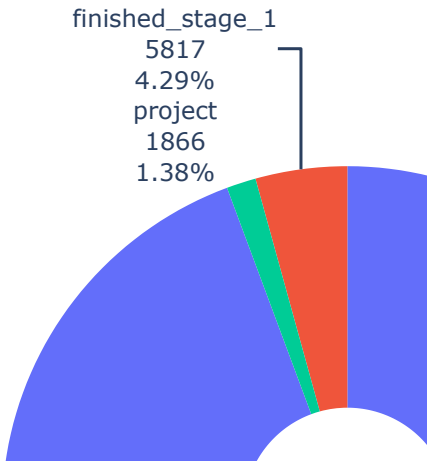


график по реализованным проектам

In [43]:

```
pie = df.groupby('project_type')['user_id'].count().reset_index()

labels = pie.project_type
values = pie.user_id

fig = go.Figure(data=[go.Pie(labels=labels, values=values,hole=0.3)])
fig.update_traces(hoverinfo='label+percent', textinfo='label+value+percent'
                  )
fig.update_layout(
    title_text="Реализованные проекты")
fig.show()
```

построить график по дням, по которому произошел клик по объявлению

In [44]:
```
import plotly.express as px
```

In [45]:
```
ad_costs.head()
```

Out[45]:

	source	day	cost
0	facebook_ads	2020-05-03	935.882786
1	facebook_ads	2020-05-04	548.354480
2	facebook_ads	2020-05-05	260.185754
3	facebook_ads	2020-05-06	177.982200
4	facebook_ads	2020-05-07	111.766796

In [46]:
```
fig = px.line(ad_costs.groupby('day')['cost'].sum().reset_index(), x='day', y='cost')
fig.update_layout(
    title_text="график по дням, по которому произошел клик по объявлению (cost)")
fig.show()
```

график для источников, с которых пришел пользователь

In [47]:
```
ad_costs
```

Out[47]:

	source	day	cost
0	facebook_ads	2020-05-03	935.882786
1	facebook_ads	2020-05-04	548.354480
2	facebook_ads	2020-05-05	260.185754
3	facebook_ads	2020-05-06	177.982200
4	facebook_ads	2020-05-07	111.766796
5	facebook_ads	2020-05-08	68.009276
6	facebook_ads	2020-05-09	38.723350
7	instagram_new_adverts	2020-05-03	943.204717
8	instagram_new_adverts	2020-05-04	502.925451
9	instagram_new_adverts	2020-05-05	313.970984
10	instagram_new_adverts	2020-05-06	173.071145
11	instagram_new_adverts	2020-05-07	109.915254
12	instagram_new_adverts	2020-05-08	71.578739
13	instagram_new_adverts	2020-05-09	46.775400
14	yandex_direct	2020-05-03	969.139394
15	yandex_direct	2020-05-04	554.651494
16	yandex_direct	2020-05-05	308.232990
17	yandex_direct	2020-05-06	180.917099
18	yandex_direct	2020-05-07	114.429338
19	yandex_direct	2020-05-08	62.961630
20	yandex_direct	2020-05-09	42.779505
21	youtube_channel_reklama	2020-05-03	454.224943
22	youtube_channel_reklama	2020-05-04	259.073224
23	youtube_channel_reklama	2020-05-05	147.041741

	source	day	cost
24	youtube_channel_reklama	2020-05-06	88.506074
25	youtube_channel_reklama	2020-05-07	55.740645
26	youtube_channel_reklama	2020-05-08	40.217907
27	youtube_channel_reklama	2020-05-09	23.314669

In [48]:

```
pie = user_source.groupby('source')['user_id'].count().reset_index()

labels = pie.source
values = pie.user_id

fig = go.Figure(data=[go.Pie(labels=labels, values=values,hole=0.3)])
fig.update_traces(hoverinfo='label+percent', textinfo='label+value+percent'
                  )
fig.update_layout(
    title_text="Переходы")
fig.show()
```

пользователи которые пришли из разных источников

Статистические гипотезы

- Проверьте гипотезу различия времени прохождения уровня между пользователями, которые заканчивают уровень через реализацию проекта, и пользователями, которые заканчивают уровень победой над другим игроком.

Сформулируйте и проверьте статистическую гипотезу относительно представленных данных:

- Проверить различие кто больше приносит денег по кликам - пользователи, которые заканчивают уровень "побеждая врага" или пользователи, которые заканчивают уровень через реализацию проекта

- H0 - время прохождения уровня между пользователями, которые заканчивают уровень через реализацию проекта, и пользователями, которые заканчивают уровень победой над другим игроком статистически значима
- H1 - различие не статистически значимо

In [49]:
```
min_event = game_actions.groupby(['user_id','event'])['time'].min().reset_index()
time_event = game_actions.query("event == 'finished_stage_1'")[['user_id','time']]
event_time = min_event.merge(time_event,on = 'user_id', how = 'inner')
event_time.columns = ['user_id','event','start','finish']
event_time['diff_time_event'] = event_time['finish'] - event_time['start']
event_time = event_time.merge(df1, on ='user_id', how = 'inner')
event_time.head(1)
```

Out[49]:

	user_id	event_x	start	finish	diff_time_event	event_datetime	event_y	building_type	project_type	time	count
0	001d39dc-366c-4021-9604-	building	2020-05-05	2020-05-12	6 days 10:38:42	2020-05-12 07:40:47	finished_stage_1	unknown	unknown	2020-05-12	warrior

	user_id	event_x	start	finish	diff_time_event	event_datetime	event_y	building_type	project_type	time	count
	6a3b9ff01e25		21:02:05	07:40:47						07:40:47	

In [50]:
```
alpha = 0.05
results = st.mannwhitneyu(event_time[event_time['count']=='warrior']['diff_time_event'], event_time[event_time['count']=='builder'
pvalue = results.pvalue
print('p-значение: ', pvalue)
if (pvalue < alpha):
    print("Отвергаем H0: разница статистически значима")
else:
    print("Не получилось отвергнуть H0: вывод о различии сделать нельзя")
```

```
p-значение:  2.9185930689384226e-15
Отвергаем H0: разница статистически значима
```

In [51]:
```
df1[df1['count']=='warrior']['event_datetime'].head()
```

Out[51]:
```
6659     2020-05-04 19:47:29
13134    2020-05-05 13:22:09
15274    2020-05-05 18:54:37
16284    2020-05-05 21:27:29
19650    2020-05-06 06:02:22
Name: event_datetime, dtype: object
```

In [52]:
```
df1[df1['count']=='warrior']['time'].head()
```

Out[52]:
```
6659    2020-05-04 19:47:29
13134   2020-05-05 13:22:09
15274   2020-05-05 18:54:37
16284   2020-05-05 21:27:29
19650   2020-05-06 06:02:22
Name: time, dtype: datetime64[ns]
```

In [53]:
```
df2 = df[df['source']=='yandex_direct'].set_index('day')['2020-05-03':'2020-05-09'].nunique()
```

In [54]:
```
df2
```

Out[54]:
```
event               3
building_type       4
user_id          4728
project_type        2
time            21872
source              1
week                1
dtype: int64
```

In [55]:
```
df3 = df[df['source']=='youtube_channel_reklama'].set_index('day')['2020-05-03':'2020-05-09'].nunique()
df3
```

Out[55]:
```
event               3
building_type       4
user_id          2630
project_type        2
time            12196
source              1
week                1
dtype: int64
```

In [56]:
```
count_warrior_youtube = df.query("event == 'finished_stage_1' and source =='youtube_channel_reklama'").count()
count_warrior_youtube.unique()
```

Out[56]:
```
array([1159])
```

In [57]:
```
count_warrior_yandex = df.query("event == 'finished_stage_1' and source =='yandex_direct'").count()
count_warrior_yandex.unique()
```

Out[57]:
```
array([2042])
```

In [58]:
```
ad_costs['cost'] = ad_costs['cost'].astype(int)
```

In [59]:
```
yandex_direct = ad_costs['cost'][ad_costs['source']=='yandex_direct'].sum()
youtube_channel_reklama = ad_costs['cost'][ad_costs['source']=='youtube_channel_reklama'].sum()
yandex_direct
```

Out[59]:
```
2229
```

- H0 - нет различий между теми кто прошел уровень придя из yandex_direct и youtube_channel_reklama соответственно
- H1 - группы прошедших уровень, которые установили приложения через yandex_direct и youtube_channel_reklama разные

In [60]:
```
alpha=0.05
purchases = np.array([1159,2042])
leads = np.array([2630, 4728])
p1 = purchases[0] / leads[0]
p2 = purchases[1] / leads[1]
combined = (purchases[0] + purchases[1]) / (leads[0] + leads[1])
difference = p1-p2
z_value = difference / math.sqrt(combined * (1 - combined) * (1 / leads[0] + 1 / leads[1]))
distr = st.norm(0,1)
p_value = (1 - distr.cdf(abs(z_value))) * 2
print('p-значение: ', p_value)
if (p_value < alpha):
    print("Отвергаем нулевую гипотезу")
else:
    print("Не получилось отвергнуть нулевую гипотезу")
```

```
p-значение:  0.46611328936689755
Не получилось отвергнуть нулевую гипотезу
```

Выводы

- больше всего приходят из яндекс директа
- малое количество игроко (1%) доходят до постройки орбитальной станции
- посещения в игру к концу обозреваемого периода падают
- Ни одна гипотеза не потвердилась - различия между yandex_direct и youtube_channel_reklama нет, а также время прохождения уровня между пользователями, которые заканчивают уровень через реализацию проекта, и пользователями, которые заканчивают уровень победой над другим игроком не различается
- из графиков можно сделать вывод что интерес к игре начинает теряться на 8ой день, почти полностью теряется на 21ый день, поэтому предлагаю начинать запускать рекламу в период с 3его по 5ый день использования игры

In [64]:
```
df1.to_csv(r'C:\Users\vneso\Downloads\file.csv')
```

In [66]:
```
sudo apt-get install pandoc
```

```
  File "/tmp/ipykernel_2375/1901338604.py", line 1
    sudo apt-get install pandoc
         ^
SyntaxError: invalid syntax
```

In [ ]: