

Игры — Формирование модели монетизации

ссылка на презентацию

<https://yadi.sk/i/eirneYEZ9ruerw>

ссылка на дашборд

<https://public.tableau.com/profile/tatiana.pugacheva#!/>

Описание игры

Название - «Космические братья». В ней пользователи строят свою космическую программу и пытаются преуспеть в нелёгком деле колонизации галактики.

Основная монетизация игры — только планируется. Но предполагается, что в приложении будет происходить показ рекламы на экране с выбором типа объекта для постройки.

Задача

Сформировать модель монетизации "Космических братьев".

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.

Нужно выбрать оптимальное время для запуска рекламы. Зная расходы на продвижение игры, можно расчитать её окупаемость при разных сценариях

Пока создатели игры планируют показывать её на экране выбором постройки.

Данные

В нашем распоряжении 3 датасета

Датасет *game_actions.csv*:

- *event_datetime* — время события;
- *event* — одно из трёх событий:
 1. *building* — объект построен,
 2. *finished_stage_1* — первый уровень завершён,
 3. *project* — проект завершён;
- *building_type* — один из трёх типов здания:
 1. *assembly_shop* — сборочный цех,

2. `spaceport` — космопорт,
 3. `research_center` — исследовательский центр;
- `user_id` — идентификатор пользователя;
 - `project_type` — тип реализованного проекта;

Помимо основного датасета есть два датасета с информацией о рекламных активностях. Они также помогут в решении задачи.

Датасет `ad_cost.csv` содержит колонки:

- `day` - день, в который был совершен клик по объявлению
- `source` - источник трафика
- `cost` - стоимость кликов

Датасет `user_source.csv` содержит колонки:

- `user_id` - идентификатор пользователя
- `source` - источников, с которого пришёл пользователь, установивший приложение

Желаемые результаты

Построить модель, при которой пользователи будут продолжать играть, обращая внимание на рекламу (при этом не раздражаясь)

План работы

1. Открытие файла. Изучение общей информации

Посмотрим, какими данными мы располагаем. Сделаем первые выводы (количество строк, пропущенных значений, типы данных)

1. Предобработка данных

Исходя из выводов, сделанных на первом шаге, "почистим" данные, если это необходимо

1. Исследовательский анализ данных

Посмотрим на распределения и взаимные корреляции и выявим какие-то аномалии, если они есть; сформулируем первые гипотезы

1. Анализ влияния событий на совершение целевого события

Целевых событий два - победа над врагом или реализация проекта - разработка орбитальной сборки спутников. Посмотрим, что чаще выбирают пользователи и как другие события влияют на результат

1. Проверка гипотез

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

1. Выбор оптимального времени для показа рекламы

Посмотрим, на каком этапе у нас остается достаточное количество пользователей, чтобы показать рекламу, не уходя при этом в минус. Мы знаем, сколько средств было потрачено на рекламу нашей игры. По этим данным расчитаем сумму, необходимую для неухода в минус

1. Tableau

Планирую сделать дашборд

1. Вывод

Общий вывод из проделанной работы, рекомендации для монетизации игры

Открытие файла. Изучение общей информации

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from plotly import graph_objects as go
import plotly.express as px
from plotly.graph_objs import *
import plotly.io as pio
pio.templates
from sklearn.preprocessing import LabelEncoder
from scipy import stats as st
import numpy as np
import math as mth
from datetime import datetime
```

```
In [3]: ad_costs = pd.read_csv('ad_costs.csv')
```

```
In [4]: game_actions = pd.read_csv('game_actions.csv')
```

```
In [5]: user_source = pd.read_csv('user_source.csv')
```

```
In [6]: ad_costs
```

```
Out[6]:
```

	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

	source	day	cost
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
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 [7]: `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 [8]: `game_actions.head()`

	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 [9]: `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 [10]: `user_source.head()`

	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 [11]: `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 [12]: `game_actions.duplicated().sum()`

Out[12]: 1

In [13]: `user_source.duplicated().sum()`

Out[13]: 0

У нас есть 3 датасета

`game_actions :`

- 135640 строк
- пропущенные значения в столбце с выбранным проектом и с типом здания.
Появились из-за того, что некоторые пользователи не выбирали проект (постройку здания). А значит, заменим пропуски в этих строках на `no_building` в типах знания и `in_process` - если игрок еще не закончил постройку, `enemy_defeated` - если игрок победил врага, в столбце с выбранным проектом
- нужно заменить тип данных в столбце с датами
- есть один дубликат, избавимся от него

user_source :

- 13676 строк
- преобразований типов данный не требуется
- нужно объединить таблицу с таблицей ad_costs для дальнейшей работы
- пропусков нет

ad_costs

- 28 строк
- пропусков нет
- требуется преобразование типов данных в столбце с датой

Предобработка данных

```
In [14]: game_actions = game_actions.drop_duplicates().reset_index(drop=True)
```

```
In [15]: game_actions['event_datetime'] = pd.to_datetime(game_actions['event_datetime'])
```

```
In [16]: ad_costs['day'] = pd.to_datetime(ad_costs['day'])
```

```
In [17]: game_actions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 135639 entries, 0 to 135638
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   event_datetime  135639 non-null   datetime64[ns]
 1   event           135639 non-null   object  
 2   building_type   127956 non-null   object  
 3   user_id          135639 non-null   object  
 4   project_type    1866 non-null    object  
 dtypes: datetime64[ns](1), object(4)
memory usage: 5.2+ MB
```

```
In [18]: game_actions['building_type'].unique()
```

```
Out[18]: array(['assembly_shop', 'spaceport', nan, 'research_center'], dtype=object)
```

```
In [19]: game_actions['building_type'] = game_actions['building_type'].fillna('no_building')
```

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

```
Out[20]: array([nan, 'satellite_orbital_assembly'], dtype=object)
```

```
In [21]: game_actions['event'].unique()
```

```
Out[21]: array(['building', 'finished_stage_1', 'project'], dtype=object)
```

```
In [22]: tab_for_filling = game_actions.pivot_table(index='user_id', values = 'project_type',
```

```
In [23]: tab_for_filling.loc[tab_for_filling['project_type']!=0]['user_id']
```

```
Out[23]: 13      006380d6-67a6-45de-ab1e-6eb54ad8dbe8
24      00be7371-fa79-4fbdb8e7-8ba32e3f8f82
26      00c2b9f2-2615-439c-aed0-6974e4c6c1c1
```

```

33      00de2809-403e-4476-a2f7-ca1b11700190
36      00e83261-5fd4-4939-9616-802b5a0202c6
...
13539   ff50f63a-c4b3-494e-8bf9-3f19dbfc513e
13542   ff60fcda-4329-494e-b82d-461fbdf5b5eb
13549   ff83bdd8-f49a-49fd-8aad-9fa1fb30d516
13550   ff841992-d93b-462e-bc6e-18c24a514f36
13575   ffff69cc-fec1-4fd3-9f98-93be1112a6b8
Name: user_id, Length: 1866, dtype: object

```

```
In [24]: game_actions['project_type'] = game_actions['project_type'].fillna('in_process')
```

```
In [25]: def filling_project(row):
    if (row['event'] == 'finished_stage_1') and (row['user_id'] not in tab_for_fill)
        row['project_type'] = 'enemy_defeated'
    return row['project_type']
```

```
In [26]: game_actions['project_type'] = game_actions.apply(filling_project, axis = 1)
```

```
In [27]: game_actions['day'] = game_actions['event_datetime'].astype('datetime64[D]')
```

```
In [28]: user_source = user_source.merge(game_actions.groupby('user_id').agg({'day':'min'})).r
```

Так как рекламу начали 3 мая, а игру запустили 4 мая, сдвинем даты в таблице ad_costs на 1 день, чтобы корректно посчитать затраты на пользователя

```
In [29]: game_actions.head()
```

```
Out[29]:
```

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

```
In [30]: ad_costs['day'] = ad_costs['day'] + np.timedelta64(1, 'D')
```

```
In [31]: ad_costs = ad_costs.merge(user_source.groupby(['day', 'source']).agg({'user_id':'nun
```

```
In [32]: ad_costs = ad_costs.rename(columns = {'user_id':'n_users'})
ad_costs
```

```
Out[32]:
```

	source	day	cost	n_users
0	facebook_ads	2020-05-04	935.882786	1184
1	instagram_new_adverts	2020-05-04	943.204717	1452
2	yandex_direct	2020-05-04	969.139394	2078
3	youtube_channel_reklama	2020-05-04	454.224943	1146
4	facebook_ads	2020-05-05	548.354480	694

	source	day	cost	n_users
5	instagram_new_adverts	2020-05-05	502.925451	796
6	yandex_direct	2020-05-05	554.651494	1208
7	youtube_channel_reklama	2020-05-05	259.073224	663
8	facebook_ads	2020-05-06	260.185754	345
9	instagram_new_adverts	2020-05-06	313.970984	472
10	yandex_direct	2020-05-06	308.232990	664
11	youtube_channel_reklama	2020-05-06	147.041741	366
12	facebook_ads	2020-05-07	177.982200	224
13	instagram_new_adverts	2020-05-07	173.071145	270
14	yandex_direct	2020-05-07	180.917099	395
15	youtube_channel_reklama	2020-05-07	88.506074	220
16	facebook_ads	2020-05-08	111.766796	139
17	instagram_new_adverts	2020-05-08	109.915254	175
18	yandex_direct	2020-05-08	114.429338	239
19	youtube_channel_reklama	2020-05-08	55.740645	135
20	facebook_ads	2020-05-09	68.009276	84
21	instagram_new_adverts	2020-05-09	71.578739	112
22	yandex_direct	2020-05-09	62.961630	144
23	youtube_channel_reklama	2020-05-09	40.217907	100
24	facebook_ads	2020-05-10	38.723350	56
25	instagram_new_adverts	2020-05-10	46.775400	70
26	yandex_direct	2020-05-10	42.779505	89
27	youtube_channel_reklama	2020-05-10	23.314669	56

In [33]: `ad_costs['cost'].sum()`

Out[33]: 7603.576986788023

Данные обработали, перейдем к анализу

Исследовательский анализ данных

Посмотрим, сколько игроков выбрали победить врага, а сколько построить станцию

In [34]: `game_actions.groupby('project_type').agg({'user_id':'nunique', 'event_datetime':'max'})`

Out[34]:

project_type	user_id	event_datetime
--------------	---------	----------------

project_type	user_id	event_datetime
enemy_defeated	3951	2020-06-05 12:32:49
in_process	13576	2020-06-05 08:57:52

user_id	event_datetime
project_type	
satellite_orbital_assembly	1866 2020-06-04 08:29:42

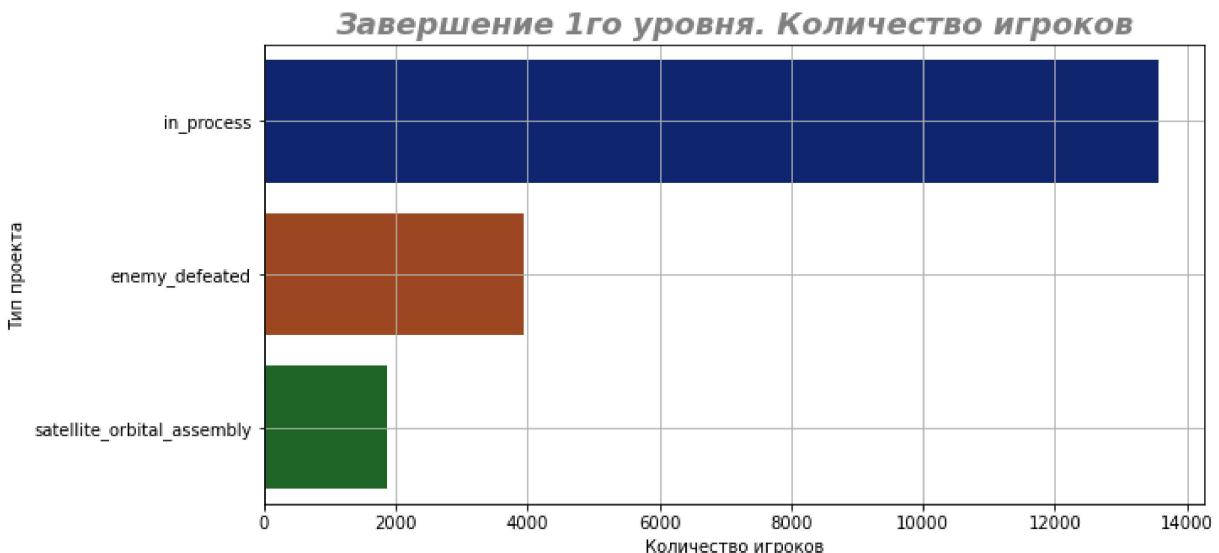
In [35]: `game_actions.user_id.nunique()`

Out[35]: 13576

```
In [36]: project_users = game_actions.groupby('project_type').agg({'user_id':'nunique'}).reset_index()
sns.set_palette('dark')
plt.figure(figsize = (10, 5))
sns.barplot(data = project_users,
             y = 'project_type', x='user_id')

plt.grid()
plt.title('Завершение 1го уровня. Количество игроков', color="grey", fontsize=18, fontweight='bold')
plt.xlabel('Количество игроков')
plt.ylabel('Тип проекта')

plt.show()
```



In [37]: `fig = go.Figure(data=[go.Pie(labels=project_users['project_type'], values=project_us fig.show()`

Заверши



Почти в три раза больше игроков победило врага, чем построили станцию. Можно сделать вывод, что победить врага проще, чем построить станцию

НО показывать рекламу планируется на этапе постройки, а большинство игроков стройку не выбирает

In [38]: `game_actions.describe()`

```
<ipython-input-38-34573facd9f5>:1: FutureWarning:
```

```
Treating datetime data as categorical rather than numeric in `describe` is deprecated and will be removed in a future version of pandas. Specify `datetime_is_numeric=True` to silence this warning and adopt the future behavior now.
```

```
<ipython-input-38-34573facd9f5>:1: FutureWarning:
```

```
Treating datetime data as categorical rather than numeric in `describe` is deprecated and will be removed in a future version of pandas. Specify `datetime_is_numeric=True` to silence this warning and adopt the future behavior now.
```

Out[38]:

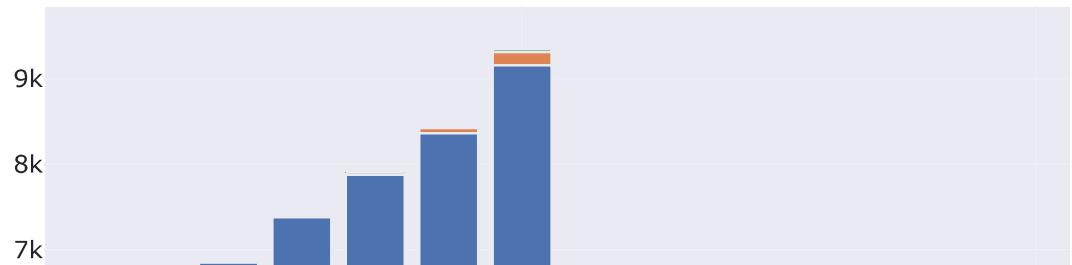
	event_datetime	event	building_type		user_id	project_type	day
count	135639	135639	135639		135639	135639	135639
unique	128790	3	4		13576	3	33
top	2020-05-09 12:35:56	building	spaceport	bf542075-e3a2-4e79-82d8-3838e86d2a25	in_process	2020-05-10 00:00:00	
freq	4	127956	59325		22	129822	15207
first	2020-05-04 00:00:01	NaN	NaN		NaN	NaN	2020-05-04 00:00:00
last	2020-06-05 12:32:49	NaN	NaN		NaN	NaN	2020-06-05 00:00:00

In [39]: `events_per_day = game_actions.pivot_table(index='day', 'event'), values = 'user_id'`

In [40]: `fig = px.bar(events_per_day, x='day', y='user_id', color='event',`

```
title='События по дням', template = 'seaborn')
fig.update_xaxes(tickangle=45)
fig.update_layout(xaxis_title='дата', yaxis_title='количество игроков')
#fig.update_layout(Layout(paper_bgcolor='rgba(0,0,0,0)', plot_bgcolor='rgba(0,0,0,0)')
```

```
fig.update_traces(marker_line_width=1)
fig.show()
```



Приложение запустили 4 мая. 5 мая первые пользователи уже победили врага. И только 8 мая перешли к проекту. Значит, победа над врагом требует меньше времени и сил.

Посмотрим на взимные корреляции

```
In [41]: encoder = LabelEncoder()
game_actions['event_num'] = encoder.fit_transform(game_actions['event'])
game_actions['building_type_num'] = encoder.fit_transform(game_actions['building_type'])
game_actions['project_type_num'] = encoder.fit_transform(game_actions['project_type'])
```

```
In [42]: game_actions.corr()
```

```
Out[42]:
```

	event_num	building_type_num	project_type_num
event_num	1.000000	-0.096130	-0.008456
building_type_num	-0.096130	1.000000	0.030988
project_type_num	-0.008456	0.030988	1.000000

Нет явный зависимостей

```
In [44]: gameplay_time = game_actions.pivot_table(index='user_id', values = 'event_datetime',
```

```
In [45]: gameplay_time.columns = ['gameplay_time']
```

```
In [46]: gameplay_time['seconds'] = gameplay_time['gameplay_time'].dt.seconds/3600
```

```
In [47]: gameplay_time['days'] = gameplay_time['gameplay_time'].dt.days * 24
```

```
In [48]: gameplay_time['gameplay_time'] = gameplay_time['seconds'] + gameplay_time['days']
```

```
In [49]: gameplay_time = gameplay_time.reset_index()
```

```
In [50]: game_actions = game_actions.merge(gameplay_time, on = 'user_id')
```

```
In [51]: game_actions
```

Out[51]:

		event_datetime	event	building_type	user_id	project_type	day	event_num	bu
0	2020-05-04 00:00:01	building	assembly_shop	55e92310- cb8e-4754- b622- 597e124b03de	in_process	2020- 05-04		0	
1	2020-05-05 05:20:15	building	assembly_shop	55e92310- cb8e-4754- b622- 597e124b03de	in_process	2020- 05-05		0	
2	2020-05-05 19:24:02	building	assembly_shop	55e92310- cb8e-4754- b622- 597e124b03de	in_process	2020- 05-05		0	
3	2020-05-05 20:59:19	building	assembly_shop	55e92310- cb8e-4754- b622- 597e124b03de	in_process	2020- 05-05		0	
4	2020-05-06 00:52:56	building	assembly_shop	55e92310- cb8e-4754- b622- 597e124b03de	in_process	2020- 05-06		0	
...
135634	2020-05-13 23:56:39	building	spaceport	cafd1b8b- 507b-49ae- abf0- 89fe863a8b03	in_process	2020- 05-13		0	
135635	2020-05-18 00:23:28	building	spaceport	cafd1b8b- 507b-49ae- abf0- 89fe863a8b03	in_process	2020- 05-18		0	
135636	2020-05-10 23:59:52	building	assembly_shop	bdc53e19- 462a-4dc3- 8b55- b4a34278af6a	in_process	2020- 05-10		0	
135637	2020-05-13 08:11:19	building	spaceport	bdc53e19- 462a-4dc3- 8b55- b4a34278af6a	in_process	2020- 05-13		0	

	event_datetime	event	building_type	user_id	project_type	day	event_num	bu
135638	2020-05-18 11:11:11	building	spaceport	bdc53e19-462a-4dc3-8b55-b4a34278af6a	in_process	2020-05-18	0	

135639 rows × 12 columns

In [52]: `gameplay_time.merge(gameplay_time, on = 'user_id')`

Out[52]:

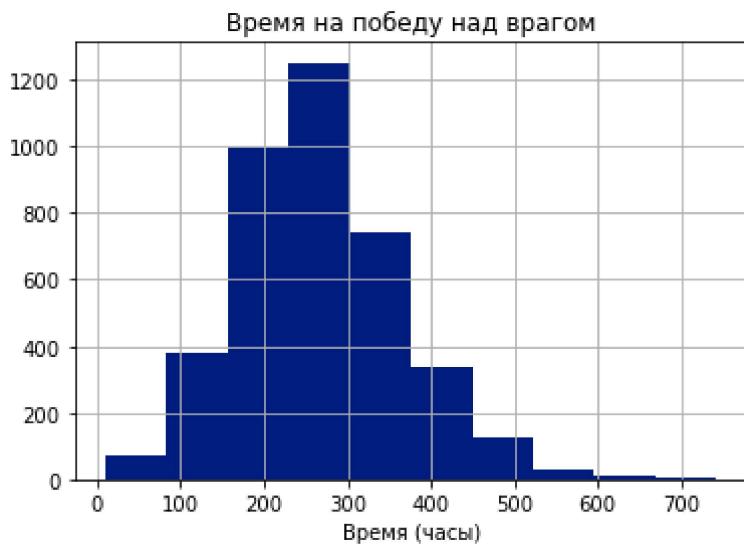
	user_id	gameplay_time_x	seconds_x	days_x	gameplay_time_y	seconds_y	days_y
0	0001f83c-c6ac-4621-b7f0-8a28b283ac30		346.308056	10.308056	336	346.308056	10.308056
1	00151b4f-ba38-44a8-a650-d7cf130a0105		295.627778	7.627778	288	295.627778	7.627778
2	001aaea6-3d14-43f1-8ca8-7f48820f17aa		207.209722	15.209722	192	207.209722	15.209722
3	001d39dc-366c-4021-9604-6a3b9ff01e25		154.645000	10.645000	144	154.645000	10.645000
4	002f508f-67b6-479f-814b-b05f00d4e995		396.946389	12.946389	384	396.946389	12.946389
...
13571	ffef4fed-164c-40e1-bde1-3980f76d0fb5		188.281944	20.281944	168	188.281944	20.281944
13572	ffffab3da-da0e-4e30-ae62-10d0a2e24a4e		238.222778	22.222778	216	238.222778	22.222778
13573	ffffb626c-5ab6-47c9-8113-2062a2f18494		202.945278	10.945278	192	202.945278	10.945278
13574	fffff194a-56b7-4c12-860d-3485242ae7f5		169.600556	1.600556	168	169.600556	1.600556
13575	fffff69cc-fec1-4fd3-9f98-93be112a6b8		415.139167	7.139167	408	415.139167	7.139167

13576 rows × 7 columns

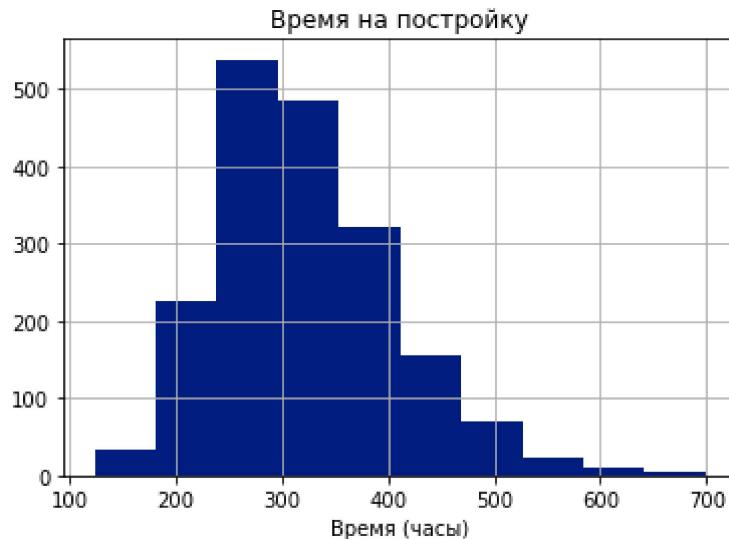
In [53]: `game_actions.head()`

	event_datetime	event	building_type	user_id	project_type	day	event_num	building
0	2020-05-04 00:00:01	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-04	0	
1	2020-05-05 05:20:15	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-05	0	
2	2020-05-05 19:24:02	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-05	0	
3	2020-05-05 20:59:19	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-05	0	
4	2020-05-06 00:52:56	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-06	0	

In [54]: `game_actions.loc[game_actions['project_type'] == 'enemy_defeated']['gameplay_time'].hist()
plt.title('Время на победу над врагом')
plt.xlabel('Время (часы)')
plt.show()`



In [55]: `game_actions.loc[game_actions['project_type'] == 'satellite_orbital_assembly']['gameplay_time'].hist()
plt.title('Время на постройку')
plt.xlabel('Время (часы)')
plt.show()`



Игроки потратили примерно одинаковое время на победу над врагом и на постройку проекта. Проверим эту гипотезу

Анализ влияния событий на совершение целевого события

```
In [56]: actions_new = pd.get_dummies(game_actions[['event', 'building_type', 'project_type',
```

```
In [57]: actions_new.head()
```

```
Out[57]:
```

	event_datetime	gameplay_time	event_building	event_finished_stage_1	event_project	building_ty
0	2020-05-04 00:00:01	164.45	1	0	0	
1	2020-05-05 05:20:15	164.45	1	0	0	
2	2020-05-05 19:24:02	164.45	1	0	0	
3	2020-05-05 20:59:19	164.45	1	0	0	
4	2020-05-06 00:52:56	164.45	1	0	0	

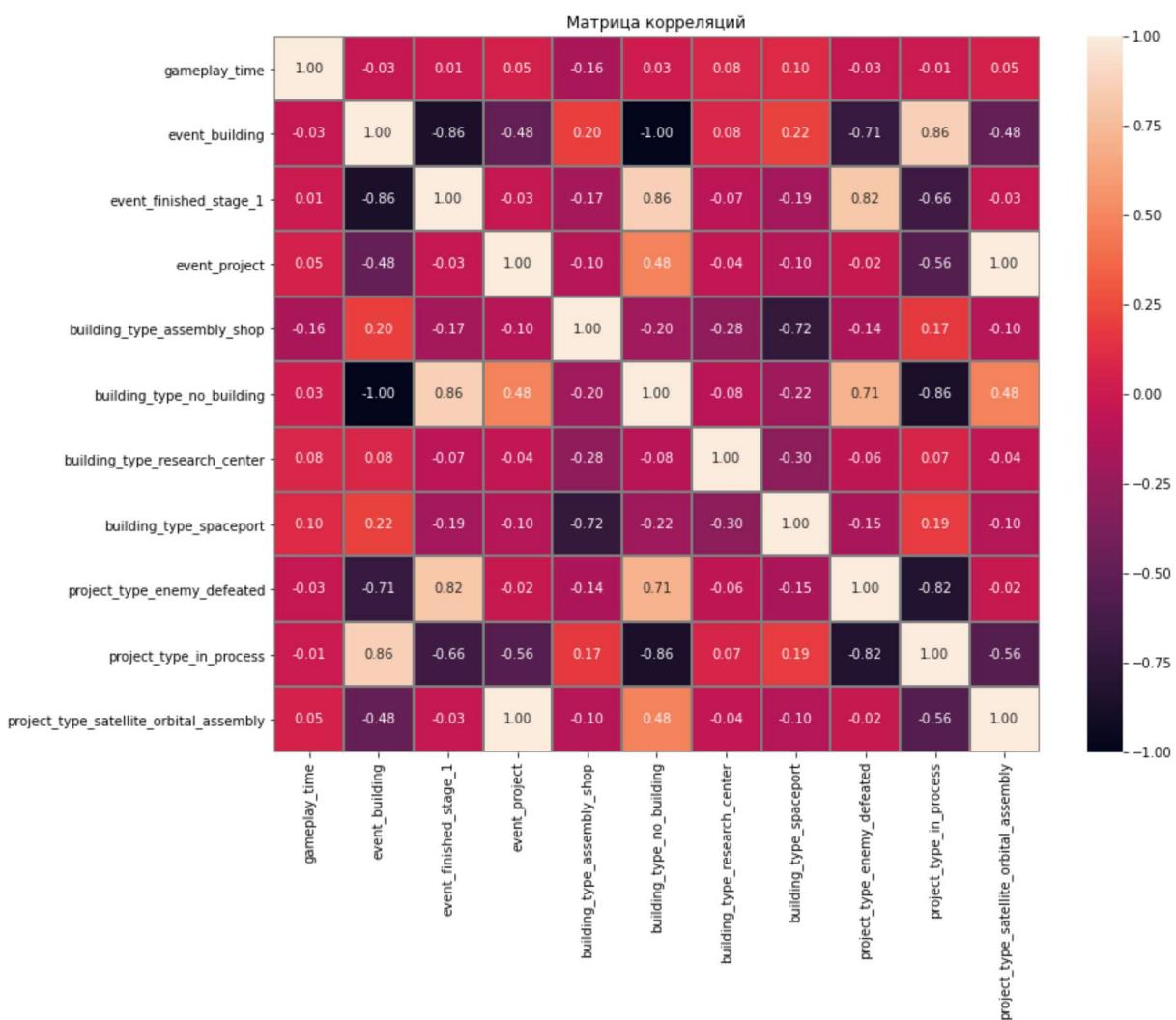
```
In [58]: actions_new.corr()
```

```
Out[58]:
```

	gameplay_time	event_building	event_finished_stage_1	event_project
gameplay_time	1.000000	-0.030582	0.006141	0.
event_building	-0.030582	1.000000	-0.863854	-0.
event_finished_stage_1	0.006141	-0.863854	1.000000	-0.
event_project	0.050009	-0.481988	-0.025000	1.
building_type_assembly_shop	-0.163293	0.200807	-0.173468	-0.
building_type_no_building	0.030582	-1.000000	0.863854	0.
building_type_research_center	0.084125	0.083584	-0.072204	-0.

	gameplay_time	event_building	event_finished_stage_1	event_project	building_type_no_building
building_type_spaceport	0.095312	0.216049	-0.186635	-0.	
project_type_enemy_defeated	-0.027240	-0.706879	0.818286	-0.	
project_type_in_process	-0.006141	0.863854	-0.664843	-0.	
project_type_satellite_orbital_assembly	0.050009	-0.481988	-0.025000	1.	

```
In [59]: plt.figure(figsize = (13, 10))
sns.heatmap(actions_new.corr(), annot=True, fmt='.2f', linewidths=1, linecolor='gray'
plt.title('Матрица корреляций')
plt.show()
```



Ожидаемо - на победу над врагом влияет выбор типа здания - без здания, на конечный проект ничто не влияет

Проверка гипотез

```
In [60]: time_pivot = game_actions.pivot_table(index = 'user_id', values = 'gameplay_time', c
```

```
In [61]: game_actions
```

```
Out[61]: event_datetime event building_type user_id project_type day event_num bu
```

	event_datetime	event	building_type	user_id	project_type	day	event_num	bu
0	2020-05-04 00:00:01	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-04	0	
1	2020-05-05 05:20:15	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-05	0	
2	2020-05-05 19:24:02	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-05	0	
3	2020-05-05 20:59:19	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-05	0	
4	2020-05-06 00:52:56	building	assembly_shop	55e92310-cb8e-4754-b622-597e124b03de	in_process	2020-05-06	0	
...
135634	2020-05-13 23:56:39	building	spaceport	cafd1b8b-507b-49ae-abf0-89fe863a8b03	in_process	2020-05-13	0	
135635	2020-05-18 00:23:28	building	spaceport	cafd1b8b-507b-49ae-abf0-89fe863a8b03	in_process	2020-05-18	0	
135636	2020-05-10 23:59:52	building	assembly_shop	bdc53e19-462a-4dc3-8b55-b4a34278af6a	in_process	2020-05-10	0	
135637	2020-05-13 08:11:19	building	spaceport	bdc53e19-462a-4dc3-8b55-b4a34278af6a	in_process	2020-05-13	0	
135638	2020-05-18 11:11:11	building	spaceport	bdc53e19-462a-4dc3-8b55-b4a34278af6a	in_process	2020-05-18	0	

135639 rows × 12 columns

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

In [62]: game_actions.groupby('project_type').count()

Out[62]:

project_type	event_datetime	event	building_type	user_id	day	event_num	buildi
project_type							
enemy_defeated	3951	3951	3951	3951	3951	3951	3951
in_process	129822	129822	129822	129822	129822	129822	129822
satellite_orbital_assembly	1866	1866	1866	1866	1866	1866	1866

```
In [63]: game_actions.pivot_table(index = 'user_id', values = 'gameplay_time', columns = 'pro
```

Out[63]:

project_type	enemy_defeated	in_process	satellite_orbital_assembly
user_id			
0001f83c-c6ac-4621-b7f0-8a28b283ac30	NaN	346.308056	NaN
00151b4f-ba38-44a8-a650-d7cf130a0105	NaN	295.627778	NaN
001aaea6-3d14-43f1-8ca8-7f48820f17aa	NaN	207.209722	NaN
001d39dc-366c-4021-9604-6a3b9ff01e25	154.645000	154.645000	NaN
002f508f-67b6-479f-814b-b05f00d4e995	NaN	396.946389	NaN
...
ffef4fed-164c-40e1-bde1-3980f76d0fb5	188.281944	188.281944	NaN
fffab3da-da0e-4e30-ae62-10d0a2e24a4e	NaN	238.222778	NaN
ffffb626c-5ab6-47c9-8113-2062a2f18494	202.945278	202.945278	NaN
ffff194a-56b7-4c12-860d-3485242ae7f5	NaN	169.600556	NaN
fffff69cc-fec1-4fd3-9f98-93be1112a6b8	NaN	415.139167	415.139167

13576 rows × 3 columns

```
In [64]: sample_1 = time_pivot.loc[time_pivot['enemy_defeated'] > 0]['enemy_defeated']
```

```
In [65]: sample_2 = time_pivot.loc[time_pivot['satellite_orbital_assembly'] > 0]['satellite_o
```

```
In [67]: print('Нулевая гипотеза - между средним временем прохождения уровням через реализации')
print('\n')
print('Среднее время прохождения уровня победой над врагом: ', sample_1.mean())
print('Среднее время прохождения уровня постройками: ', sample_2.mean())
print('\n')
alpha = .05 # критический уровень статистической значимости
# если p-value окажется меньше него - отвергнем гипотезу

results = st.ttest_ind(
    sample_1,
    sample_2)

print('p-значение:', results.pvalue)

if (results.pvalue < alpha):
    print("Отвергаем нулевую гипотезу - между средними есть разница")
else:
```

```
print("Не получилось отвергнуть нулевую гипотезу - между средними разницы нет")
```

Нулевая гипотеза - между средним временем прохождения уровня через реализацию проекта и через победу над врагом нет разницы

Среднее время прохождения уровня победой над врагом: 266.5066878286794
 Среднее время прохождения уровня постройками: 323.01364862450947

p-значение: 7.256845329498955e-98

Отвергаем нулевую гипотезу - между средними есть разница

```
In [68]: def ztest(total_a, total_b, part_a, part_b):
    # критический уровень статистической значимости

    total = np.array([total_a, total_b])
    parts = np.array([part_a,
                     part_b])

    p1 = parts[0]/total[0]

    p2 = parts[1]/total[1]

    p_combined = (parts[0] + parts[1]) / (total[0] + total[1])

    difference = p1 - p2

    # считаем статистику в ст.отклонениях стандартного нормального распределения
    z_value = difference / mth.sqrt(p_combined * (1 - p_combined) * (1/total[0] + 1))

    # задаем стандартное нормальное распределение (среднее 0, ст.отклонение 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: между долями есть значимая разница")
    else:
        print("Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли
```

ГИПОТЕЗА_2

Доля игроков, завершивших уровень победой над врагом, статистически равна доле пользователей, завершивших уровень постройкой

```
In [69]: total_all = len(game_actions['user_id'].unique())
part_enemy = len(game_actions.loc[game_actions['project_type']=='enemy_defeated']['u
part_build = len(game_actions.loc[game_actions['project_type']=='satellite_orbital_a
```

```
In [70]: print('Количество игроков: ', total_all)
print('Игроки, победившие врага: ', part_enemy)
print('Игроки, построившие станцию: ', part_build)
ztest(total_all, total_all, part_enemy, part_build)
```

Количество игроков: 13576

Игроки, победившие врага: 3951

Игроки, построившие станцию: 1866

р-значение: 0.0

Отвергаем нулевую гипотезу: между долями есть значимая разница

ГИПОТЕЗА_3

Доли пользователей, завершивших уровни, статистически равны для всех источников

```
In [71]: game_actions = game_actions.merge(user_source[['user_id', 'source']], on = 'user_id')
```

```
In [72]: alpha = alpha / 6
for i in (game_actions['source'].unique()):
    total_one = len(game_actions.loc[game_actions['source']==i]['user_id'].unique())
    part_one = len(game_actions.loc[
        (game_actions['source']==i) & (game_actions['project_type']!='in_process')])
    for j in (game_actions['source'].unique()):
        if j!=i:
            total_two = len(game_actions.loc[game_actions['source']==j]['user_id'].u
            part_two = len(game_actions.loc[
                (game_actions['source']==j) & (game_actions['project_type']!='in_process'
            print('Всего игроков из', i, total_one)
            print('Завершили уровень', part_one)

            print('Всего игроков из', j, total_two)
            print('Завершили уровень', part_two)

            print('Нулевая гипотеза: доли пользователей, завершивших уровни, статисти
ztest(total_one, total_two, part_one, part_two)
print('\n')
```

Всего игроков из youtube_channel_reklama 2686

Завершили уровень 1159

Всего игроков из facebook_ads 2726

Завершили уровень 1150

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис
точника youtube_channel_reklama и facebook_ads

р-значение: 0.4737412931629539

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из youtube_channel_reklama 2686

Завершили уровень 1159

Всего игроков из instagram_new_adverts 3347

Завершили уровень 1466

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис
точника youtube_channel_reklama и instagram_new_adverts

р-значение: 0.6123664865502236

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из youtube_channel_reklama 2686

Завершили уровень 1159

Всего игроков из yandex_direct 4817

Завершили уровень 2042

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис
точника youtube_channel_reklama и yandex_direct

р-значение: 0.5244230207169549

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из facebook_ads 2726

Завершили уровень 1150

Всего игроков из youtube_channel_reklama 2686

Завершили уровень 1159

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис

точника facebook_ads и youtube_channel_reklama

р-значение: 0.4737412931629539

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из facebook_ads 2726

Завершили уровень 1150

Всего игроков из instagram_new_adverts 3347

Завершили уровень 1466

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис

точника facebook_ads и instagram_new_adverts

р-значение: 0.20644137086490777

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из facebook_ads 2726

Завершили уровень 1150

Всего игроков из yandex_direct 4817

Завершили уровень 2042

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис

точника facebook_ads и yandex_direct

р-значение: 0.8624388026512437

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из instagram_new_adverts 3347

Завершили уровень 1466

Всего игроков из youtube_channel_reklama 2686

Завершили уровень 1159

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис

точника instagram_new_adverts и youtube_channel_reklama

р-значение: 0.6123664865502236

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из instagram_new_adverts 3347

Завершили уровень 1466

Всего игроков из facebook_ads 2726

Завершили уровень 1150

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис

точника instagram_new_adverts и facebook_ads

р-значение: 0.20644137086490777

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из instagram_new_adverts 3347

Завершили уровень 1466

Всего игроков из yandex_direct 4817

Завершили уровень 2042

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис

точника instagram_new_adverts и yandex_direct

р-значение: 0.20595754565724533

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из yandex_direct 4817

Завершили уровень 2042

Всего игроков из youtube_channel_reklama 2686

Завершили уровень 1159

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для ис

точника yandex_direct и youtube_channel_reklama

р-значение: 0.5244230207169549

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из yandex_direct 4817

Завершили уровень 2042

Всего игроков из facebook_ads 2726

Завершили уровень 1150

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для источника `yandex_direct` и `facebook_ads`

p-значение: 0.8624388026512437

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Всего игроков из `yandex_direct` 4817

Завершили уровень 2042

Всего игроков из `instagram_new_adverts` 3347

Завершили уровень 1466

Нулевая гипотеза: доли пользователей, завершивших уровни, статистически равны для источника `yandex_direct` и `instagram_new_adverts`

p-значение: 0.20595754565724533

Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

Выбор оптимального времени показа рекламы

Посмотрим, сколько пользователей прошло через каждый этап

```
In [73]: game_actions.groupby('event').agg({'user_id':'count'})
```

```
Out[73]:      user_id
```

	event
building	127956
finished_stage_1	5817
project	1866

```
In [74]: ad_costs
```

```
Out[74]:      source        day    cost  n_users
```

0	facebook_ads	2020-05-04	935.882786	1184
1	instagram_new_adverts	2020-05-04	943.204717	1452
2	yandex_direct	2020-05-04	969.139394	2078
3	youtube_channel_reklama	2020-05-04	454.224943	1146
4	facebook_ads	2020-05-05	548.354480	694
5	instagram_new_adverts	2020-05-05	502.925451	796
6	yandex_direct	2020-05-05	554.651494	1208
7	youtube_channel_reklama	2020-05-05	259.073224	663
8	facebook_ads	2020-05-06	260.185754	345
9	instagram_new_adverts	2020-05-06	313.970984	472
10	yandex_direct	2020-05-06	308.232990	664
11	youtube_channel_reklama	2020-05-06	147.041741	366
12	facebook_ads	2020-05-07	177.982200	224
13	instagram_new_adverts	2020-05-07	173.071145	270
14	yandex_direct	2020-05-07	180.917099	395
15	youtube_channel_reklama	2020-05-07	88.506074	220

	source	day	cost	n_users
16	facebook_ads	2020-05-08	111.766796	139
17	instagram_new_adverts	2020-05-08	109.915254	175
18	yandex_direct	2020-05-08	114.429338	239
19	youtube_channel_reklama	2020-05-08	55.740645	135
20	facebook_ads	2020-05-09	68.009276	84
21	instagram_new_adverts	2020-05-09	71.578739	112
22	yandex_direct	2020-05-09	62.961630	144
23	youtube_channel_reklama	2020-05-09	40.217907	100
24	facebook_ads	2020-05-10	38.723350	56
25	instagram_new_adverts	2020-05-10	46.775400	70
26	yandex_direct	2020-05-10	42.779505	89
27	youtube_channel_reklama	2020-05-10	23.314669	56

Расчитаем стоимость клика на игрока

```
In [75]: ad_costs['cost_per_user'] = ad_costs['cost'] / ad_costs['n_users']
```

Сколько всего было потрачено на рекламу

```
In [76]: ad_costs['cost'].sum()
```

```
Out[76]: 7603.576986788023
```

```
In [77]: n_users = game_actions.groupby('event').agg({'user_id':'count'})
```

```
In [78]: n_users
```

```
Out[78]:
```

event	user_id
building	127956
finished_stage_1	5817
project	1866

Сколько в среднем потратили на пользователя

```
In [79]: ad_costs['cost'].sum()/ad_costs['n_users'].sum()
```

```
Out[79]: 0.5600749106355349
```

Посчитаем стоимость кликов из каждого источника

```
In [80]: ad_costs.groupby('source').agg({'cost_per_user':'mean', 'n_users':'sum'})
```

```
Out[80]:
```

source	cost_per_user	n_users
facebook_ads	0.776358	2726

source	cost_per_user	n_users
instagram_new_adverts	0.646144	3347
yandex_direct	0.463491	4817
youtube_channel_reklama	0.403225	2686

У ютуба самая низная цена клика, но яндекс привел больше всего игроков, немножко превышая в цене ютуб

Готова третья гипотеза. Доли пользователей, завершивших первый уровень, из разных источников равны

Если каждый ролик показывать по стоимости клика, то

```
In [81]: n_users['cost'] = n_users['user_id'] * ad_costs['cost'].sum() / ad_costs['n_users'].su
```

```
In [82]: n_users = n_users.reset_index()
n_users
```

```
Out[82]:
```

	event	user_id	cost
0	building	127956	71664.945265
1	finished_stage_1	5817	3257.955755
2	project	1866	1045.099783

Получается, мы полностью окупимся, если показывать ролик по стоимости клика на экране выбора здания. Но это самая первая ступень, и игроки чаще выбирают победу над врагом

Поэтому предлагаю такой вариант показа рекламы - после завершения уровня

```
In [83]: print('Сумма за ролик для выхода в 0 - ', ad_costs['cost'].sum() / (n_users.loc[n_us
```

Сумма за ролик для выхода в 0 - 0.9896624999073308

Соответственно, для выхода в плюс, стоимость ролика должна быть больше 0,99

Каждый игрок смотрит рекламу один раз - после завершения уровня

Вывод

- игроки чаще выбирают победу над врагом
- среднее время для прохождения уровня одинаково для любого выбора игрока
- монетизация приложения на экране выбора постройки - плохая идея:
 - игрок еще не втянулся в игру
 - игроки часто не выбирают постройки вообще
 - оптимальный вариант - показ рекламы после завершения уровня. Так можно выйти в 0, даже в плюс. Игроки не будут раздражаться, так как показ рекламы только после прохождения уровня
- для постоянного источника трафика подходит яндекс директ - привел больше всего пользователей, СРС - на втором месте