

Table of Contents

- ▼ [1 Introduction](#)
 - [1.1 What is GodsUnchained?](#)
 - [1.2 What have I tried to do up to this point?](#)
 - [2 Let's see if we can work with ERC721 tokens via Etherscan API](#)
 - [3 Get transaction history for a contract](#)
 - [4 Store transactions in a DataFrame](#)
- ▼ [5 Token Name category](#)
 - [5.1 Trying to match this address to ERC-721 transactions](#)

1 Introduction

1.1 What is GodsUnchained?

- Gods Unchained is a card game that prides itself on players having "true ownership" of its in-game assets. One of the first claims on its website is that you can use this to build extremely rare decks and sell them if you want to; a feature most modern games are adamantly against
 - cards are associated with ERC-721 tokens - each is unique and some are rarer than others
- GU is built by ex-Google and Riot Games engineers
- cards unlocked just by playing the game

1.2 What have I tried to do up to this point?

- In the other notebook I handed in, I just worked on using Python to get transaction details and lists of transactions for a given Ethereum wallet. In this notebook, I'm doing the same thing with ERC-721 tokens.
- In this notebook, I get the transaction history of one contract (one that deals with a lot, if not all, GU transactions) as well as the transaction history of what I believe to be the address that all GU transactions go through
- Later on in the notebook, I try to see the ratio between the number of transactions to the GU wallet address vs. the number that are sent elsewhere - I'm assuming anything that isn't going to this GU address must be coming from it, since that's what we searched for.
- I'm thinking that any ERC-721 token can be traced back to a GU transaction if it is part of this contract address (discussed more in depth later) or if its to/from the wallet address

2 Let's see if we can work with ERC721 tokens via Etherscan API

<https://etherscan.io/apis#accounts> (<https://etherscan.io/apis#accounts>) choose ERC721 tokens towards the bottom on this link

choose a GodsUnchained card holder from here:

https://bloxy.info/token_holders/0x0e3a2a1f2146d86a604adc220b4967a898d7fe07
(https://bloxy.info/token_holders/0x0e3a2a1f2146d86a604adc220b4967a898d7fe07)

For this example, we'll use the following address:
 0x26cdee4269273e1ea5dfac6b5791df2656897738

Finding the right address was a little trickier for this one. First, go to the etherscan page for your chosen wallet address (wallet must have a transaction with ERC721 tokens):
<https://etherscan.io/address/0x26cdee4269273e1ea5dfac6b5791df2656897738#tokentxnsErc721>
<https://etherscan.io/address/0x26cdee4269273e1ea5dfac6b5791df2656897738#tokentxnsErc721>)

First, go over to the ERC721 tokens tab and click on whatever contract you want the transactions for.

Address 0x26CDee4269273E1ea5DFAC6b5791Df2656897738

Feature Tip: \$ DEFI - Track your Compound & Maker loans on Etherscan!

Overview

Balance: 0.010185162768860833 Ether

Ether Value: \$1.90 (@ \$186.96/ETH)

Token: \$0.00 14

More Info

My Name Tag: Not Available, Update?

Transactions Erc20 Token Txns **Erc721 Token Txns** Analytics Comments

Latest 25 ERC-721 Token Transfer Events

| Txn Hash | Age | From | To | Token ID | Token |
|-----------------------------|--------------------|-----------------------|------------------------|----------|-----------------------------|
| 0xa49013adecdf8cc... | 11 days 23 hrs ago | 0x0000000000000000... | IN 0x26cdee4269273e... | 100 | GU: Genesis ... (GU:G:B...) |

Look for the 'Contract' address on this page - this is what we'll be using to search

Status: Success

Block: 9969774 77208 Block Confirmations

Timestamp: 11 days 23 hrs ago (Apr-29-2020 08:56:26 PM +UTC)

From: 0x2ceeb12a9b5419bc29efffe5a13880a4f587f844

To: **Contract 0x5c1749bc5734b8f9ea7cda7e38b47432c6c6b66**

Tokens Transferred: (40 ERC-721 Transfers found)

- From 0x0000000000000000... To 0x721931508df2764... For ERC-721 TokenID [81] GU: Genesis ... (GU:G:B...)
- From 0x0000000000000000... To 0x721931508df2764... For ERC-721 TokenID [82] GU: Genesis ... (GU:G:B...)
- From 0x0000000000000000... To 0x721931508df2764... For ERC-721 TokenID [83] GU: Genesis ... (GU:G:B...)

```
In [2]: ▶ import requests # the requests module will help us connect to etherscan

address = '0x5c1749bc5734b8f9ea7cda7e38b47432c6cffb66' # contract address
key = 'MV3PBAW3Y5U9IC3WQKZGX8NTUGS9VAEZ28' # my api key
# change https to http (not encrypted)

url = 'https://api.etherscan.io/api?module=account&action=tokennfttx&contract=' + address + '&page=1&offset=100&sort=desc&apikey=' + key

"""
NOTE - in the url:
    offset changes the number of records that will be searched (in this case,
    sort can change whether you're looking at the most recent transactions or
"""

# connect to address
response = requests.get(url)
print(response)
```

<Response [200]>

Response 200 means we're getting a valid result

3 Get transaction history for a contract

```
In [3]: ▶ address_content = response.json()
result = address_content.get("result") # get only the result of the transaction
#print(result)
```

```
In [ ]: ▶ # for transaction in result:
#         print(transaction)
```

4 Store transactions in a DataFrame

```
In [4]: ▶ import pandas as pd
test_df = pd.DataFrame(columns=['ID', 'Hash', 'From', 'To', 'Value', 'contractAddress', 'tokenName', 'Confirmations'])
test_df
```

Out[4]:

| ID | Hash | From | To | Value | contractAddress | tokenName | Confirmations |
|----|------|------|----|-------|-----------------|-----------|---------------|
|----|------|------|----|-------|-----------------|-----------|---------------|

```
In [5]: # Lets try to get data from individual transactions
# we'll get some additional data with these transactions
for n, transaction in enumerate(result):
    # this will loop through and get the data we want from each record and th
    hash_ = transaction.get("hash")
    tx_from = transaction.get("from")
    tx_to = transaction.get("to")
    value = transaction.get("value")
    contractAddress = transaction.get("contractAddress")
    tokenName = transaction.get("tokenName")
    confirmations = transaction.get("confirmations")

    #new_data = [n, hash_, tx_from, tx_to, value, confirmations]
    test_df = test_df.append({'ID': n, 'Hash': hash_, 'From': tx_from, 'To':

test_df
```

Out[5]:

| | ID | | Hash |
|---|----|---|-----------------------------|
| 0 | 0 | 0xbebe0f236b10558c6b53c04eafb2d4d88420876fee7... | 0x27be4acb4e615b3f5e9b171eb |
| 1 | 1 | 0x9b6e1adc00089a24738f078840264497a39cc0835bb6... | 0x4ef40d1bf0983899892946830 |
| 2 | 2 | 0x9b6e1adc00089a24738f078840264497a39cc0835bb6... | 0xbc9ca5bd0f07700929f8d5382 |
| 3 | 3 | 0x22951da6b6183f26093157500f06f75a6039c5bb85a1... | 0x4ef40d1bf0983899892946830 |
| 4 | 4 | 0x22951da6b6183f26093157500f06f75a6039c5bb85a1... | 0x24618c0d57fd54e55aacc5830 |

5 Token Name category

I'm interested in the tokenName category, so I looked into it a bit more. It seems like this token has its own contract that can be explored more in depth:

Token GU: Genesis Board ⓘ

Buy Exchange Crypto Credit

Feature Tip: Track historical data points of any address with the [analytics module](#) !

Overview [ERC-721]

Total Supply: 6,042 GU:G:BRD ⓘ

Holders: 3,581 addresses

Transfers: 6,281

Profile Summary [Edit]

Contract: [0x5c1749bc5734b8f9ea7cda7e38b47432c6cffb66](#)

Social Profiles: Not Available, [Update ?](#)

Transfers Holders Inventory Info DEX Trades Read Contract Write Contract Comments

Token Holders Chart

Top 1,000 holders (From a total of 3,581 holders)

First < Page 1 of 20 > Last

| Rank | Address | Quantity | Percentage |
|------|--|----------|------------|
| 1 | 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2 | 46 | 0.7613% |
| 2 | 0x4c55c92ee0937aaf21eb087c92739204daa2d58 | 28 | 0.4634% |

- Transfers
 - shows transactions under this contract
- Holders
 - shows the addresses of wallets that currently hold these tokens
 - this is the page the screenshot above is showing - the address we searched for in the last notebook holds 0.7613% of all of these tokens
- Inventory
 - I believe this is showing each token and the wallet address that owns it

I'm still trying to make complete sense of what this is. Here is the link for what I'm taking about, if you want to poke around:

[https://etherscan.io/token/0x5c1749bc5734b8f9ea7cda7e38b47432c6cffb66?
a=0x721931508df2764fd4f70c53da646cb8aed16ace](https://etherscan.io/token/0x5c1749bc5734b8f9ea7cda7e38b47432c6cffb66?a=0x721931508df2764fd4f70c53da646cb8aed16ace)
([https://etherscan.io/token/0x5c1749bc5734b8f9ea7cda7e38b47432c6cffb66?
a=0x721931508df2764fd4f70c53da646cb8aed16ace](https://etherscan.io/token/0x5c1749bc5734b8f9ea7cda7e38b47432c6cffb66?a=0x721931508df2764fd4f70c53da646cb8aed16ace))

Etherscan

Eth: \$186.90 (-0.69%)

Home Blockchain Tokens Resources More ABZDEL

All Filters Search by Address / Txn Hash / Block / Token / Ens

Token GU: Genesis Board ⓘ

Buy Exchange Crypto Credit

Feature Tip: Track historical data points of any address with the [analytics module](#) !

Overview [ERC-721]

Total Supply: 6,042 GU:G:BRD ⓘ

Holders: 3,581 addresses

Profile Summary [Edit]

Contract: [0x5c1749bc5734b8f9ea7cda7e38b47432c6cffb66](#)

Social Profiles: Not Available, [Update ?](#)

FILTERED BY TOKEN HOLDER

[0x721931508df2764fd4f70c53da646cb8aed16ace](#)

BALANCE

18 GU:G:BRD

My thought is that the 'filtered by token holder' is the actual wallet address that all GU transactions go through, and the contract address is the contract that encapsulates all of the transactions.

5.1 Trying to match this address to ERC-721 transactions

```
In [6]: ▶ # this url searches the above wallet for the past 1000 transactions
# address = 0x721931508DF2764fD4F70C53Da646Cb8aEd16acE
new_url = 'https://api.etherscan.io/api?module=account&action=txlist&address=
```

```
In [7]: ▶ test_df = pd.DataFrame(columns=['ID', 'Hash', 'From', 'To', 'Value', 'Confirm
test_df # clear test DataFrame
```

Out[7]:

| ID | Hash | From | To | Value | Confirmations |
|----|------|------|----|-------|---------------|
|----|------|------|----|-------|---------------|

```

In [8]: response = requests.get(new_url)
address_content = response.json()
result = address_content.get("result") # get only the result of the transaction

# Lets try to get data from individual transactions
for n, transaction in enumerate(result):
    # this will loop through and get the data we want from each record and then
    hash_ = transaction.get("hash")
    tx_from = transaction.get("from")
    tx_to = transaction.get("to")
    value = transaction.get("value")
    confirmations = transaction.get("confirmations")

    #new_data = [n, hash_, tx_from, tx_to, value, confirmations]
    test_df = test_df.append({'ID': n, 'Hash': hash_, 'From': tx_from, 'To': tx_to, 'Value': value, 'Confirmations': confirmations})

test_df

```

Out[8]:

| | ID | Hash |
|----|----|--|
| 0 | 0 | 0x6e0223152b1c86f8bb86e1d34f12eba357701d9c4b31... |
| 1 | 1 | 0x70e0ed53764894eefaeb73c504481640e9848e6e5b7d... |
| 2 | 2 | 0xa92710dc5855094cdb01defae562f8a6d0470592a53c... |
| 3 | 3 | 0x34bdd9894fed30bdda06aa1c8948d2d45f3ea44623c... |
| 4 | 4 | 0x5cf3056280d6f238dc22ecf75ac96ff775f78c74f3c5... |
| 5 | 5 | 0xd5ca8f658f498cdd76749601a945d1e55b6fc910364a... |
| 6 | 6 | 0xdc6c0b1348adf519990f6f0c33f540b7a4d02608feb8... |
| 7 | 7 | 0x368224aa1f39b276fbc1099e9e993f2511b26a460d5e... |
| 8 | 8 | 0x3a7bc255f732b085556471e87d8ef1a3a2d4815665bd... |
| 9 | 9 | 0x3da946ef50f9b13bd8719431392382f6352b3a56e3d9... |
| 10 | 10 | 0x2bdbbc71ff501f25d785477d803d9a4ea05e82bdfadd8... |
| 11 | 11 | 0x93dc94ac68b9fda6b163f0e85ea70e43c34a6389930f... |
| 12 | 12 | 0xf6b3472a05e894160856e53d29a394e2099c4efe1361... |
| 13 | 13 | 0xcd4c30fc4d1d010e6e3b1519a5276cd7557ba16eb7cd... |
| 14 | 14 | 0x561af5f4bcdd02aad16c0cfdecf067d257f00c678019... |
| 15 | 15 | 0x0b6149d1fdf51d04d8ca2d8cd6f2490e57e3035d235b... |
| 16 | 16 | 0xa30e8eafaa55e6fbf396e8e43a7337b301f018bc8cb7... |
| 17 | 17 | 0xa60d5783499e422b8c31bd79781943ffbef0a9cc192d... |
| 18 | 18 | 0xdbcd872f8ef9528fed1080188fc322a77807b5420ed3... |
| 19 | 19 | 0x4c013dcd6ad589ad25cab0c3aa0970dd905b7ff0bc51... |
| 20 | 20 | 0x71f350cc0ae65cda765419cdf10ab2a8f063d949bf60... |
| 21 | 21 | 0x48b0dc9e45e7e299bc6a5a7b341a9505f926d308d6d1... |

| | ID | Hash |
|-----|-----|--|
| 22 | 22 | 0xcba52db79104bd2d7e5c7661415198c58c567295b214... 0x721931508df2764fd4f70c53 |
| 23 | 23 | 0xdd6edd7197bf71281dab332ed8ffa09a3fa3b1ff55d7... 0x721931508df2764fd4f70c53 |
| 24 | 24 | 0xab7502e3741fa7469c70846e08c7d247193fa6285798... 0x721931508df2764fd4f70c53 |
| 25 | 25 | 0xf6550755fa230747cbc4772cd51df7b9b85d96c0d3f1... 0x721931508df2764fd4f70c53 |
| 26 | 26 | 0x3b83da86c026d3abc2b9c2537f34e128f3d9336b6029... 0x721931508df2764fd4f70c53 |
| 27 | 27 | 0xf08d0a4ef13c4ed74cacf30a63df98fac0c5464bddad... 0x721931508df2764fd4f70c53 |
| 28 | 28 | 0x95777b6c85ee2fef2550e4d68e06d9c8cb45e3cbc66b... 0x721931508df2764fd4f70c53 |
| 29 | 29 | 0x1f24f20a974ff42397f9d11b90efa058090d2d328353... 0x721931508df2764fd4f70c53 |
| ... | ... | ... |
| 970 | 970 | 0x801626bfc9f9144df341ba58e8f12b2b879df1658334... 0x721931508df2764fd4f70c53 |
| 971 | 971 | 0xc0b62e686d3118571f246a843718b1c11aa85da312e0... 0x721931508df2764fd4f70c53 |
| 972 | 972 | 0x540d5b257bec7c5c99744c06ad8d70b22e5737ac737a... 0x721931508df2764fd4f70c53 |
| 973 | 973 | 0xadebd6689224a3f033f3899f9c9bc7c190f1dac0aff8... 0x721931508df2764fd4f70c53 |
| 974 | 974 | 0xd3ddb2311ddb678ef00c455cd1c084f902613f193838... 0x721931508df2764fd4f70c53 |
| 975 | 975 | 0xfb1b3a9cbc55811b90a4526ba17e1cd6fcdc046e031a... 0x721931508df2764fd4f70c53 |
| 976 | 976 | 0xfe139f92eabd5e40b2951a4a82d68d4dbc123e6ab9d1... 0x721931508df2764fd4f70c53 |
| 977 | 977 | 0xbf4dafaf06bb81ab2556673e6dc62d6b442ababfa7c1... 0x721931508df2764fd4f70c53 |
| 978 | 978 | 0xce0516dc3c27b8eab2f824622a436320a316cc0fa871... 0x721931508df2764fd4f70c53 |
| 979 | 979 | 0x8f3550e91ccd933d4737f33b0eeb533275f4c236fb5c... 0x721931508df2764fd4f70c53 |
| 980 | 980 | 0x304e97b8062d3e828861b6bc79580f438eae5f6910d7... 0x721931508df2764fd4f70c53 |
| 981 | 981 | 0x99067ac3f34c5823c7927836f8f7ee7bf1437a7f7285... 0x721931508df2764fd4f70c53 |
| 982 | 982 | 0xa9fc476b3772bc9efa5f46ebeac78d1f3a93b9ce0418... 0x721931508df2764fd4f70c53 |
| 983 | 983 | 0xe3797a6f08f8ad9c36716a4b8387411c020ceedc623f... 0x721931508df2764fd4f70c53 |
| 984 | 984 | 0x5638af73d7db9effcd0228e1ab59f90ebee9bbe460ca... 0x721931508df2764fd4f70c53 |
| 985 | 985 | 0x51cbb7ff01c03ffe6b59145005898fb7191f37db1492... 0x721931508df2764fd4f70c53 |
| 986 | 986 | 0x00c7965758d11e0072f4bb5afc34d5349d0da393541c... 0x721931508df2764fd4f70c53 |
| 987 | 987 | 0xed6fda78fca4dd7b4c081edbdd306d958a1def088880... 0x721931508df2764fd4f70c53 |
| 988 | 988 | 0x3721ca1ec4876accbac0726613b8293de91639134b31... 0x721931508df2764fd4f70c53 |
| 989 | 989 | 0xbd371f0a64700ca8dac87075101e57ea759c7599563a... 0x721931508df2764fd4f70c53 |
| 990 | 990 | 0x5b655153b27a055661cd6da77403516effc807050e18... 0x721931508df2764fd4f70c53 |
| 991 | 991 | 0xe15904d797d5184cc3de11bdf6c569fb2b4ff3f284f... 0x721931508df2764fd4f70c53 |
| 992 | 992 | 0x77d4b71bab169e9c7cbc497d05391bd1647feea7f83a... 0x721931508df2764fd4f70c53 |
| 993 | 993 | 0x015929dfe879c685954235e7176844ba7e8888853ba7... 0xc75efe55705770db12724dea |
| 994 | 994 | 0x7145b7accbf9bcf643e0884448666723788b5ec1f09... 0x721931508df2764fd4f70c53 |
| 995 | 995 | 0xf86bd171a27b1f646d801d890a15ca23e4b362e3a4cc... 0x721931508df2764fd4f70c53 |
| 996 | 996 | 0x55d2d4f33bf76ef7a65221058ea88b4c4e8f1379f181... 0x721931508df2764fd4f70c53 |

| ID | | Hash | |
|------------|-----|---|----------------------------|
| 997 | 997 | 0x191f4e14a87f15148af8f07c2f4b6ccac5b4408b495b... | 0x721931508df2764fd4f70c53 |
| 998 | 998 | 0x114c2fec98d690d40b4265823117171133cdfd99c5f8... | 0x721931508df2764fd4f70c53 |
| 999 | 999 | 0x5ba6cc8316b45bfed236f09799102dcc642afae2b49c... | 0x721931508df2764fd4f70c53 |

1000 rows × 6 columns

In [9]: `test_df.To.value_counts()`

```
Out[9]: 0x0e3a2a1f2146d86a604adc220b4967a898d7fe07    177
0x721931508df2764fd4f70c53da646cb8aed16ace    120
0x7be8076f4ea4a4ad08075c2508e481d6c946d12b    114
0xb1f195776a62b4074a940fa135e58249e49315a3     64
0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2     58
0x4d3814d4da8083b41861dec2f45b4840e8b72d68     55
0x960c6d43199e34dfdd9022e7ebdb9d154acc1c8b     41
0x71603f59db350681a71f16c8bcac1f43837aa3af     37
0xf4158e282f2317597e31c028978c7fb7275d6fb4     35
0x080bf510fcfb18b91105470639e9561022937712     25
0xd31eb2fe0f30b9bf09d635cc70f3727d7c16cad3     25
0xfeae9fe11170fc19b6f40796fd1debfa360daacc     21
0x2e61c63e045a978b51c6517c79c2592fcfbc82cd     17
0x76481caa104b5f6bccb540dae4cefaf1c398ebee     17
0xe34bb610505541b3fceff472dc5449053b632944     13
0x2af75676692817d85121353f0d6e8e9ae6ad5576     10
0x61935cbdd02287b511119ddb11aeb42f1593b7ef      9
0xd4690a51044db77d91d7aa8f7a3a5ad5da331af0      9
0x20d4cec36528e1c4563c1bfbe3de06aba70b22b4      8
0x8a9eba6cc332fda6919efbf3b8e90d163d3435b0      7
0xe0c71f6c2fea87f2155de000e090b09f65d361df      7
0x6b357e2cbae33cfb53ae9c136829d98def4a9561      6
0x73a4e5f8f6bd304917eca7aab5892e8507ac1ab9      5
0xd387a6e4e84a6c86bd90c158c6028a58cc8ac459      5
0xa3ac9844514b96bb502627ca9dceb57c4be289e3      4
0xf4985070ce32b6b1994329df787d1acc9a2dd9e2      4
0x46de7cba5086aece37bb88744639b02953f39c3d      3
0x3f86f18322a888d9b3adef38f127c941bccc014d      3
0x9c534adaa5c07382372c6e8b4cf7915c9881b277      3
0x946902d2f2c72448a3254c71d5c6726aa45b68ce      3
...
0xe568f1805fb21242d39d624f72a9a96723f700c7      1
0x1a8244ed31ffed7b1de5e0a28b4e0540b7f77f0a      1
0xf1caf10c5fe421660e8fdf2c0cdaa0be68633b3e      1
0x841e54526835b674392e2b8f0ec19be0577add87      1
0xf1a97c249aeaefc24e184ee1b96842bafb1941a5      1
0x244dbc0146014af6af8c80fd8db9b20dc5a27542      1
0xb39abb9a0ec3e334d369bcb4ec954b2bf6963e8d      1
0x4a371980faf6369552c7cdb6a81463e63fa5ac98      1
0xde3100cc4e0873530c68cfe158d8be506253f8e3      1
0x7a6425c9b3f5521bfa5d71df710a2fb80508319b      1
0xd880f27a3c4233f0371f879cf3d29cb106c5aa1e      1
0x9255ed258dcbafe165dccd93ea0a72f4ed88e24d      1
0xfaafdc07907ff5120a76b34b731b278c38d6043c      1
0x95129f1456981212ad76b9d35ec579d673d27b21      1
0x1aec8f11a7e78dc22477e91ed924fab46e3a88fd      1
0x3bd7b7fb8056cc1bce16d6d008936bd7af0cbe25      1
0xdf672e535d2eb5ab2b5a03f1ce0a3985a915ad34      1
0x330e9f4ab3d9ebf982217887544c18a279f84a37      1
0xceb8d29d8f73a0be355261bd2477367a265b3ecf      1
0xe824b96280df5c869d87e37cc531bbeb4d0c3c4f      1
0xa3615870889abef4de70ceb5b068e86912615a04      1
0xd02181fe0542f9a52bae87daf3f71caac63f8f8f      1
0x21fdc1baecfec9664190d0c814d7b6723e8b231d      1
```

```

0x3e569ecc7de9e14edd9c155b2c81485fb265a58c    1
0x5492e0112e49e5fed0a7b1278c76df655e509a99    1
0x744d70fdb2ba4cf95131626614a1763df805b9e     1
0xc1db2095c9da948699e67fa5949b89490a74e330    1
0xd9bf2905182dfa950764e0923f4e18288856fed4    1
0xdd44ae09f23b5e7872c37ae3d23eeb140ba6c097    1
0xcb832de0c1f2501fbd19ee48f801d3d729fd4d04    1
Name: To, Length: 103, dtype: int64

```

177 of the past transactions have gone to one address

```

In [10]:  def to_GU(x):
           if x==('0x721931508df2764fd4f70c53da646cb8aed16ace'):
               return 1
           else:
               return 0

```

```

In [11]:  # create new column for whether or not the transactions is to GU or from GU
           # 1 if to GU, 0 if not
           test_df['ToGU'] = test_df['To'].apply(to_GU)
           test_df.head()

```

```

Out[11]:

```

| | ID | Hash |
|---|----|---|
| 0 | 0 | 0x6e0223152b1c86f8bb86e1d34f12eba357701d9c4b31... 0x721931508df2764fd4f70c53da646cb8; |
| 1 | 1 | 0x70e0ed53764894eefaeb73c504481640e9848e6e5b7d... 0x721931508df2764fd4f70c53da646cb8; |
| 2 | 2 | 0xa92710dc5855094cdb01defae562f8a6d0470592a53c... 0x721931508df2764fd4f70c53da646cb8; |
| 3 | 3 | 0x34bdd9894fef30bdda06aa1c8948d2d45f3ea44623c... 0x721931508df2764fd4f70c53da646cb8; |
| 4 | 4 | 0x5cf3056280d6f238dc22ecf75ac96ff775f78c74f3c5... 0x721931508df2764fd4f70c53da646cb8; |

```

In [21]:  test_df.ToGU.value_counts()

```

```

Out[21]:  0    880
           1    120
           Name: ToGU, dtype: int64

```

I'm thinking that any transaction to the address 0x721931508df2764fd4f70c53da646cb8aed16ace is being sent, or maybe sold back, to GodsUnchained. Anything not going to that address must be coming from it (I think), so these are transactions from GU to other addresses, or items that a player bought in-game

```

In [ ]:

```

