MTGO card prize analyser

Tomasz Łacny

February 1st, 2017

Abstract

The main goal of this project was to create an application that would analyse data from online HTML website. The analysis should provide suggestions on what card currently should be bought, from whom and to whom it should be sold, depending on the prices on the market. In order to make as great profit as it is possible.

1 Introduction

MTGO – Magic The Gathering Online is online version of trading card game MTG. There exists a client application in which players can trade cards for another cards or virtual money (called *TIX* or *Ticket*) [1]. Parallelly there is a website on which anyone can check actual prices of each card in different shops (which are player-like bots that doesn't play, only sell/buy cards) [2]. These bots are analysing the prices on market and adjust their prices, but they do it in different time and frequency. Which leads to a situations when card can be bought for a lower price and then sold to another bot for higher price. The developed application should scan all the market and output information about such situations (but not always such a situation is favourable, which will be explained later).

2 MTGO card market

2.1 Cards

Every card in MTGO is an tradable and playable element of the game. Each card has at least two versions of it – normal and foil (main difference is in price and design) [1].

There are around 23 000 cards in MTGO which together with foil versions give us 46 000 tradable elements to analyse [2].

Cards have no fixed prices, simply if someone wants to sell a card its price must be competitive to the prices from other sellers, otherwise there would exist a high risk, that no one would buy it.

Cards are categorized in *Sets*. Same card in different Set can have different design, price and quantity on the market.

2.2 Shops (bots)

Cards can be traded with other players or with bots. Bots are shops which have some fixed prices of cards that are updated from time to time. Trading with bots is specific, the easiest way is to explain it on example.

Consider that player has 6 TIX. Player wants to buy card called *Elspeth, Sun's Champion THS* from a bot. Bot wants to sell that card for a price of 4.41 TIX. Player needs to request a trade from a bot, pay 5 TIX to get a card. The missing 0.59 TIX is stored in bots memory as a *Credit* and could be used as a currency in future transactions but only with the same bot, or bot from the same chain of bots. Consider that bot wants to buy some card at price of 0.71 and Player has that card and wants to sell it. During trade, bot will take the card and add 0.71 to the stored Credit, resulting in 1,3 Credit. Then Player can collect 1 TIX from bot and leave the rest stored in bots memory.

The chains of bots, are independent shops from which Player can buy (or sell to them) cards. Bots in chains can have different quantities of card, different prices, but they share *Credit*.

3 Developed application

3.1 Tools used

The tools used during development of this project were Visual Basic (as programming language), Excel 2016 (with Visual Basic IDE), Internet Explorer (as a tool for gathering source code of website), Magic The Gathering Online client (application in which it is possible to trade cards) and website https://www.mtgowikiprice.com/ (as a source of card prices) [3].

3.2 Interface

On Figure 1 we can see interface of provided application.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Card nan	Selling price	Selling bot name	Quant	Buying pr	Buying bot name	Earning per one card	Whole earning	-	I Get (t	Time	Tickets You have	-	Just did		
2	Keepsake Gorgon THS	0,002	cardware	1	0,003	iwingamesbot1	0,001	0,001	-	0	15.12.2016 23:56	7	-	Sowing Salt Foil BOK	Start	
3	Polukranos, World Eater THS	0,1	narmielbulk	4	0,141	vrtstorebuybot	0,041	0,164	-	-1	16.12.2016 00:29					
4	Anger of the Gods Foil THS	1,8	sbena_bot	2	2,14	mtgocardmarket 1	0,34	0,68	-	0	16.12.2016 00:29					
5	Anger of the Gods Foil THS	1,8	sbena_bot	2	2,14	mtgocardmarket 1	0,34	0,68	-	0	16.12.2016 00:35					
6	Sea God's Revenge THS	0,002	cardware	1	0,003	iwingamesbot1	0,001	0,001	-	0	16.12.2016 00:45					
7	Prowler's Helm THS	0,002	cardware	1	0,003	iwingamesbot1	0,001	0,001	-	0	16.12.2016 00:56					
8	Greenwarden of Murasa BFZ	0,13	aztecstorebuyer	3	0,2	plethorabulk	0,07	0,21	-	1	30.01.2017 18:24					

Figure 1 Interface of provided application

Each time provided algorithm finds an favourable trade it prints out information (1 - 10) like in the following table:
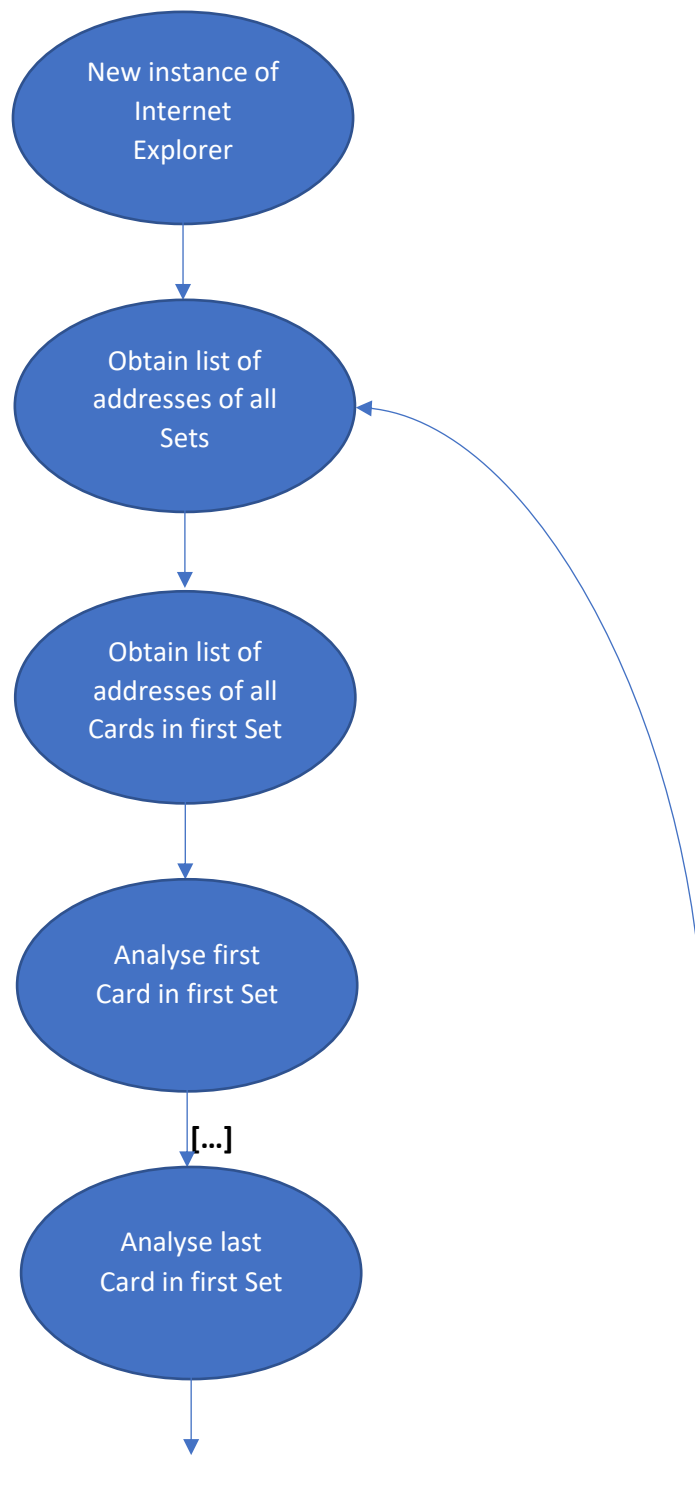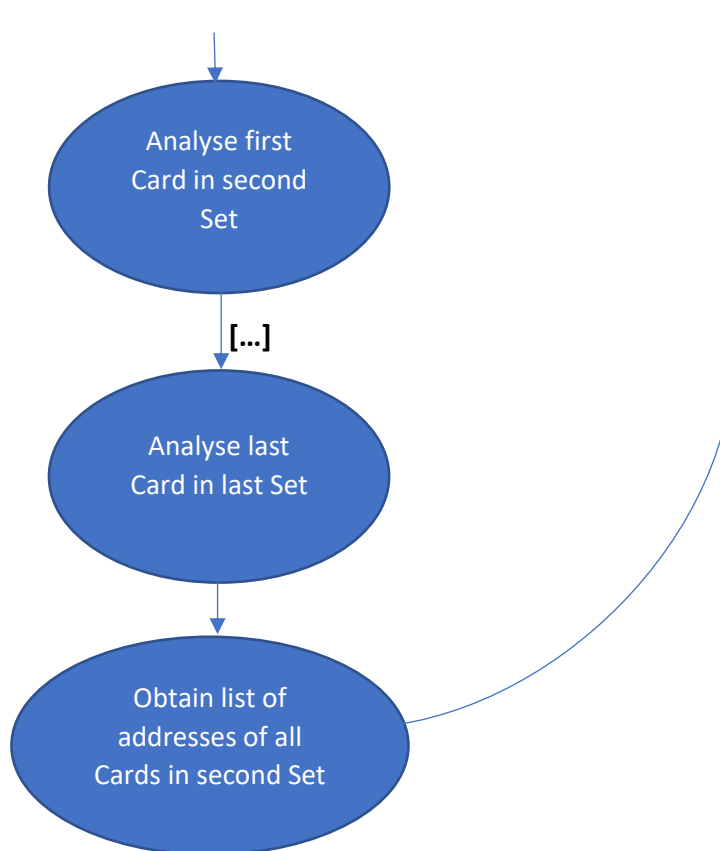
Tab. 1 Printed information

No	Column	Description
1	A	Card name
2	B	Price of single card at which Player should buy the card
3	C	Name of the bot from which Player should buy the card
4	D	Quantity of cards player should trade
5	E	Price of single card at which Player should sell the card
6	F	Name of the bot to which Player should sell the card
7	G	Profit per single card unit
8	H	Profit from whole transaction
9	J	The amount of TIX player will get after transaction
10	K	Date and time at which the favourable transaction was found

| 11 | L | Input cell for how many TIX Player wants to trade with (set manually before starting the analysis) |
| 12 | N | Name of the card that was last scanned by the algorithm (printed after every successful scan regardless of the results) |

On Figure 1 there is also a button *Start* after clicking it the application will start analysing cards, the information about newest found transaction will always be in the last non-empty row of Excel Worksheet. During analysis, the button changes its caption to *Stop*, after pressing it the algorithm will stop searching for new transactions.

3.3 Algorithm

New instance of Internet Explorer

Obtain list of addresses of all Sets

Obtain list of addresses of all Cards in first Set

Analyse first Card in first Set

[…]

Analyse last Card in first Set

During the card analysis, algorithm is searching for the best pair of seller – buyer, it considers:

1. Highest profit (some bots may have better prices but they will want lower amount of cards, so sometimes it is better to trade for lower income per card but with bigger amount of cards)
2. Highest remaining Credit (algorithm searches for maximizing remaining Credit at seller/buyer bot, which will eventually lead to condition 3)
3. Possibility of cashing Credit into operating TIX that Player could use it at other bots
4. Investing funds at bots regardless of remaining Credit if profit is higher than some fixed factor of profitability (this condition allows algorithm to find very attractive transactions with bots at which Player has no Credit)

Importance of these conditions is: 4 > 3 > 2 > 1.

4 Tests and results

The developed application was tested with factor of profitability = 0.1 TIX and 15 operating TIX (used for trades). Application was running for 10 hours 45 minutes and 23 seconds constantly, resulting in finding transactions that would provide 68,733 TIX income, but most of the income would be frozen at different bots, which leads to conclusion that factor of profitability should be significantly higher than 0.1 TIX.

The application should be tested more with different factors of profitability to obtain optimal one.

If there would have been a possibility for the application to automatize whole transaction (buying and selling) after founding attractive one, the algorithm would work significantly better.

5 Summary

The goal of the project was to develop application/algorithm that will search through data and find the most profitable seller – buyer pair and it was achieved. From tests it looks like the application is working, but effects of this work may be still improved.

References and useful sources

[1] 01.02.2017 https://www.mtgowikiprice.com

[2] 01.02.2017 http://magic.wizards.com/en/content/magic-online-products-game-info

[3] 01.02.2017 http://www.jstor.org/stable/117047?seq=1#page_scan_tab_contents

[4] 01.02.2017 http://onlinelibrary.wiley.com/doi/10.1111/j.1756-2171.2006.tb00012.x/full