

Marshall Weidman

MBA 614 Final Project Write Up

Goal

The initial goal of this project was to create a system for automating the buying and selling of items on a public market. Rather than use real markets and real money, I decided to cut my teeth on the in-game market found in the MMORPG called The Kingdom of Loathing.

Specifically, I wanted the tool to pull historical data from two different sources and display the data in a useful way. Then I wanted to pick a single item and set up a buying trigger that could run continuously, buying the specified item when the price is low enough.

Dashboard Layout

Log In

Pick an Item to Evaluate:

Inventory	Store	Store Price:
dry noodles Qty: <input type="button" value="↻"/>	dry noodles Qty: <input type="button" value="↻"/>	dry noodles 1,000

ofLoathing Historical Costs

15 Day Average	724
15 Day Median	740
30 Day Average	628
30 Day Median	738

ColdFront Market Data

Today's Qty Sold	6732
Today's Avg	720
PPU Difference from Yesterday	-18
Percent Change	-2.00%

Price

Date	Price	15 Day Mov Avg
4/11/2010	700	700
4/9/2010	500	700
4/7/2010	700	700
4/5/2010	700	700
4/3/2010	700	700
4/1/2010	700	700
3/30/2010	700	700
3/28/2010	500	650
3/26/2010	400	600
3/24/2010	100	550
3/22/2010	300	500
3/20/2010	700	500
3/18/2010	700	500
3/16/2010	700	500
3/14/2010	700	500

Current Mall Pricing

Item	Qty:	Limit:	Cost:
s Pizza Shanty & Noodle Lean-To	391	1 / day	700 Meat
Kriag's Lithp	49	2 / day	720 Meat
General's STORE shop!	27		757 Meat
The Crimson Corporation	41		758 Meat
obstradamus: the Philobstropher	91		758 Meat

Buying Trigger

Item	Buy Price:	Buy Qty:	Ownership Cap:
dry noodles	500	5	30

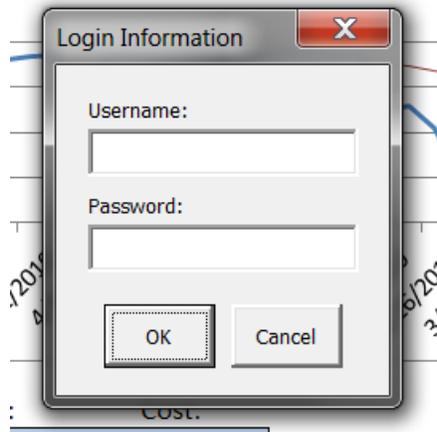
The layout above illustrates how the dashboard is set up. Starting at the top, the login button will initiate the log in sequence at the game site. Once logged in, Inventory Qty, Store Qty, Store Price, and Current Mall Pricing can all be updated by clicking the refresh button that looks like ↻.

The graph depicting the last 30 days of pricing is based on data obtained from items.ofloathing.org. This data can be updated from the dashboard. Also, another pricing database is housed at kol.coldfront.org, which provides another set of data that can be used to setting up the buying trigger.

Finally at the bottom, the buying trigger parameters need to be input based on the evaluation of the historical data provided. The simplest analysis involves setting the buy price some price lower than the 15 day median price and buying a sufficient quantity to ensure a profit upon resell.

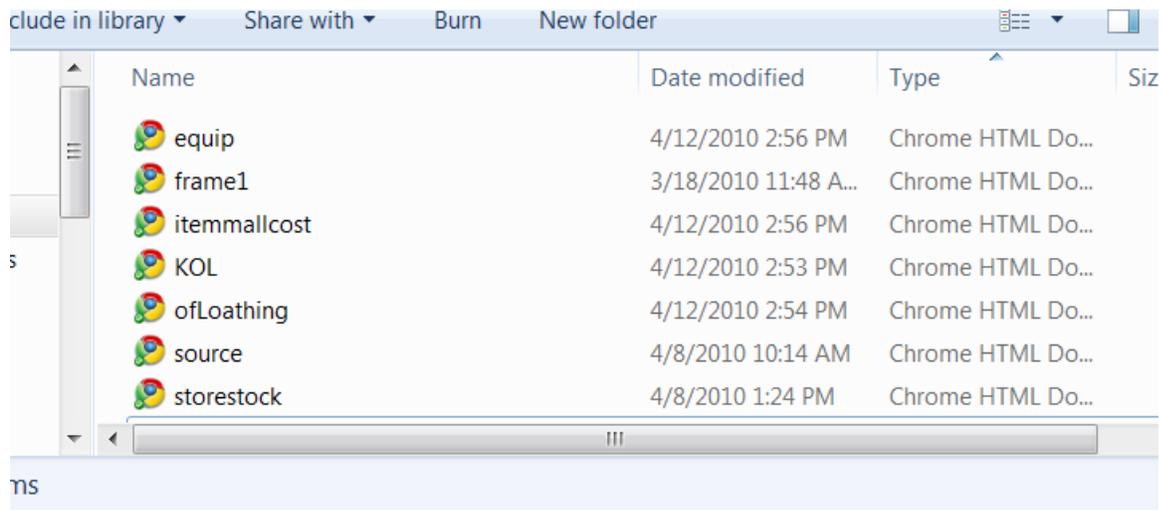
Login

Form



The form above prompts the user to input his/her username and password. This data is held only temporarily for security reasons.

Local Query's



Due to the way the website authentication is set up, direct web queries were not possible. As a work-around, once logged in, the vba is programmed to save a local copy of the HTML file and the web query actually references this local HTML instead of the live website. One hiccup I came across involves the HTML having a line that checks to ensure that it is running in a frame. I had to parse out that line of code prior to initiating the query.

Items

Inventory / Store Data

```
'click to misc
ie.document.frames(2).document.Links(1).Click
WaitForLoad

'parse out the frame check line in the html through an array then back to a file
Dim file3 As Variant
file3 = Split(ie.document.frames(2).document.all(0).outerHTML, vbNewLine)
file3(1) = ""
SaveFile ThisWorkbook.path & "\equip.html", Join(file3, vbNewLine)
'Sheets("InventoryQuery").Cells.Clear
'update the query from the local copy of the html
'UpdateQuery "URL;file:/// " & Replace(Replace(ThisWorkbook.path, "\", "/"), " ", "%20") & "/equip.html"
ActiveWorkbook.Connections("InventoryRefresh").Refresh
'add query data to the inventory page
Sheets("InventoryQuery").Activate
Sheets("InventoryQuery").Range("A1", Range("A65536").End(xlUp).Offset(10, 2).Address).Copy
Sheets("InventoryData").Range("A65536").End(xlUp).Offset(10, 0).PasteSpecial Paste:=xlPasteValues
Application.CutCopyMode = False

Call ParseInventory
Sheets(Tempsheet).Activate
Application.ScreenUpdating = True
```

As described in the previous section, local queries were used to pull the user's inventory. The inventory was actually located on 3 different pages, each one requiring a separate web query. Once all of the data was together on the spreadsheet, various parsing had to be done to remove non-required information from the inventory list. This code is referenced below.

```
'filter out stuff [, Accessory:, etc...
Sheets("InventoryData").Activate
For Each cell In Sheets("InventoryData").Range(Range("A1"), Range("A65536").End(xlUp).Address)
    lresult = Left(cell, 1)
    rresult = Right(cell, 1)
    If rresult = ":" Then
        cell.Value = ""
    ElseIf lresult = "[" Then
        cell.Value = ""
    ElseIf cell.Value = "? " Then
        cell.Value = ""
        'Place quantity of items in next cell
    ElseIf rresult = ")" Then
        OpenP = Application.WorksheetFunction.Find("(", cell, 1)
        CloseP = Application.WorksheetFunction.Find(")", cell, 1)
        tempqty = Mid(cell, OpenP + 1, CloseP - OpenP - 1)
        cell.Offset(0, 1).Value = tempqty
        cell.Value = Application.WorksheetFunction.Replace(cell.Value, OpenP - 1, CloseP, "")
    Else
        cell.Offset(0, 1).Value = 1
    End If
Next cell
```

Once the data was cleanly parsed and contained in a named range, the dashboard could effectively reference this information with the following vlookup formula.

```
=IF(ISERROR(VLOOKUP(B3,InventoryRange,2,0))=TRUE,"",VLOOKUP(B3,InventoryRange,2,0))
```

Decision Making Tools

ofLoathing Historical Costs

[p://items.ofloathing.org/](http://items.ofloathing.org/)



The Items of Loathing
An ofLoathing resource is me!



[Home](#) :: [Exchange Rate](#) :: [Oldest to Newest](#) :: [Top 200 Prices](#) :: [Most Viewed](#)

dry noodles - 737Meat

Search:

Search

About us

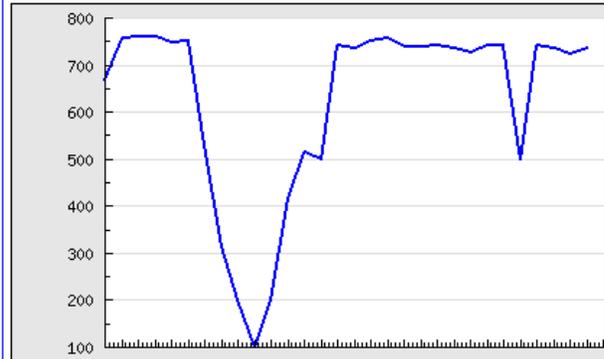
Welcome to The Items of Loathing Database! We offer a free, fast way to view item market history, trends and predictions.

If you have any suggestions, please kMail Zecious Babaloni (#694663) in-game.

Items:

dry noodles [wiki]: 737 Meat 
Use [kBay](#) for quick, easy sales.

Tomorrow: 736



[1 Month](#) :: [3 Months](#) :: [6 Months](#) :: [1 Year](#) :: [All Time](#)

Price	Date
737	04-13-10
724	04-12-10
734	04-11-10
742	04-10-10
500	04-09-10
743	04-08-10
743	04-07-10

One set of historical costs was housed at the site above. When refreshing the data, the system had to utilize the search box and couldn't just go to a specific URL. The system would then click on the '1 Year' link to get an appropriately large data set. As mentioned before, this html was saved locally and a web query was used to import the data. After parsing, the data was used to provide the dashboard analysis and graph.

ColdFront Market Data

The next source of historical costs was from kol.coldfront.org. Pulling data from this site was much more straightforward. A simple web query could be used and minimal parsing was required after the fact.

A FANSITE IS NOT!

Login

ws
at
ntact Us
mmunity Links
rums
lendar
icial Site
e Kol. Wiki

isses
ments
ims
cations
on Phases
ests
its
iat's a Wang? (FAQs)

atars
rketplace
e Day Ascension
cle scottmale24's
azing webcomic

dio Kol Broadcasts
yer Interviews
yer-created
e Official Store of
athing
dia Reviews &
erreviews

itorials

ldfront Main
rums

y of Heroes
y of Villains
e Online
posed
ldWars
eRPG
ates
ker
iW Hunters

[Ads by Google](#)
[Zouvačka Kol](#)
[Bur Kol Meat](#)
[Kol Tickets](#)
[Kol Profiling](#)

all 140 (↑ 31) | 2-ball 120 (↓ -4) | 3-ball 132 (↑ 27) | 30669 scroll 1,629 (↑ 138) | 33398 s

Last Updated: April 13th, 2010 11:45 CST
Total Meat Spent Today: **1,488,382,912**
Total Items Purchased: 186,443

Full item list

Most traded items

Item	Units Sold	PPU	Median
twinkly wad	21,647	460	6,788
dry noodles	14,200	746	2,100
scrumptious reagent	9,867	1,134	4,780
phonics down	3,473	255	1,358
pr0n legs	3,066	262	1,000

Most expensive items

Item	Units Sold	PPU	Median
Counterclockwise Watch	1	318,000,000	318,000,000
Talisman of Baio	1	77,000,000	77,000,000
navel ring of navel gazing	1	49,750,000	49,750,000
V for Vivala mask	1	38,500,000	38,500,000
Apathargic Bandersnatch	1	33,900,000	33,900,000

Greatest Price Increase

Item	Today's PPU	Difference from Yesterday	Median
Counterclockwise Watch	318,000,000	318,000,000	318,000,000
Talisman of Baio	77,000,000	77,000,000	77,000,000
navel ring of navel gazing	49,750,000	49,750,000	49,750,000
Libram of Candy Heart	17,800,000	17,800,000	17,800,000
Summoning	17,800,000	17,800,000	17,800,000
sane hatrack	7,900,000	7,900,000	7,900,000

Greatest Price Decrease

Item	Today's PPU	Difference from Yesterday	Median
Elvish sunglasses	3,690,000	-480,000	3,690,000
Travoltan trousers	9,000,000	-450,000	9,000,000
haiku katana	8,487,500	-295,833	8,487,500
depleted Grimacite grappling hook	300,000	-30,024	300,000
depleted Grimacite gravy boat	242,500	-20,250	242,500

Greatest Percentage Increase

Item	Today's PPU	% Change	Median
tiny bindle	7,998	7,848	7,998
goth kid t-shirt	7,605	7,395	7,605
Mer-kin killscroll	9,999	9,651	9,999
meat paste	30,318	27,599	10,000
towel	120,410	93,561	300,000

Greatest Percent Decrease

Item	Today's PPU	% Change	Median
cranberries	107	-1,637	450
ion-pulse modulation stabilizer	700	-9,300	700
sake bomb	300	-2,100	750
yeti fur	30	-154	30
Ankh of Badahnkadh	400	-1,595	400

Current Mall Pricing

While the two previous historical price lists only needed to be updated daily, the current mall pricing was in constant need of updating. This critical information played a major role in the user's ability to effectively make buying decisions.



rooder
level 13
Walrus
Jdgeoner

205 (170)
117 (110)
155 (110)

97 / 197

200

inture:
[y's Back 40](#)

ar:
ney, the 30
vous Tick

ts:
ng Frenzy (3)
ie Madrina

Search the Mail:

Search for : in All Categories

[\[show advanced search options\]](#)

Search Results: (Items 1-1 of 1)

dry noodles

Store:	Stock:	Limit:	Price:
Dave's Pizza Shanty & Noodle Lean-To	294	1 / day	700 Meat [buy] [buy some]
Scarydave's Scaryshop	2	1 / day	725 Meat [buy] [buy some]
The Elite Phoenix Bar	1		740 Meat [buy] [buy some]
ChromeDome 'n' Wizardry	1,276	5 / day	758 Meat [buy] [buy some]
Welcome to Rapture	6	3 / day	760 Meat [buy] [buy some]
Bill's Shop in the Mall	11	1 / day	760 Meat [buy] [buy some]
Manju's Yard	51	1 / day	770 Meat [buy] [buy some]
Dogfish Head Discounts	10	3 / day	774 Meat [buy] [buy some]
Lobstradamus: the Philobstropher	20		774 Meat [buy] [buy some]
Hugs n Bugs	14	5 / day	775 Meat [buy] [buy some]
Badde Boutique	119	5 / day	775 Meat [buy] [buy some]
Whoville General Store	5		775 Meat [buy] [buy some]
Head on the Door	53		777 Meat [buy] [buy some]
Vinny's Value Vortex	375	30 / day	777 Meat [buy] [buy some]
Chemistry Lab	97	3 / day	777 Meat [buy] [buy some]

Buying Trigger

Limits

The most technically challenging part of the project involved programming the system to buy items according to the user's specified parameters. This was especially difficult because, as seen in the graphic above, many sellers limit the quantity/per day available to each player. Once the limit has been reached, the player may no longer buy from this seller. This was difficult because once the limit was reached, the links to buy were still available, and the only way to tell if the limit had been met was to evaluate the color of the text. If the text was greyed out, then the store needed to be skipped in the buying routine. The logic below was used to asses each store and determine how much should be purchased.

If TempLimit < Qty And TempStock > Qty And Not GreyedOut = True Then

If TempLimit = 999999 And TempStock < Qty Then

If TempLimit < Qty And TempStock < Qty And Not GreyedOut = True Then

If TempLimit = 999999 And TempStock > Qty Then

If the purchase passed one of the conditions below, then the purchase was made, and logged for future reference.

Ownership Cap

Finally, the last thing that I wanted to happen was for the user to accidentally purchase more than was intended. For that reason, I created an ownership cap parameter. After each time through the 'buy trigger loop', the system would update the user's inventory and check to make sure that the ownership cap had not yet been reached.

```
For x = 1 To 25
  DoEvents
  If StopBuyLoop = True Then Exit For
  If Sheets("Dashboard").Range("InvQtyItem1").Value = "" Then
    CurrInv = 0
  Else
    CurrInv = Sheets("Dashboard").Range("InvQtyItem1").Value
  End If

  If (CurrInv + CurrStock + PurchaseCount) > ItemCap Then Exit For
  Call BuyItem(TargetItem, TargetQuantity, TargetPrice)
Next
Call UpdateInventory
```

Conclusion

This project ended up working out better than I had expected. There were a number of tricky issues to work through along the way. I learned more about HTML, Frames, and Web Queries in the last few weeks than ever before. If I have the time, I want to add a "Selling Trigger" so that I can set the buy limit and the sell limit and then let the system loop all day, buying and selling items as asking price fluctuates.