Section 1. Introduction

7EX simulates a trader's workstation, an interactive window on a dynamic securities market.

It is primarily a teaching tool, designed to give users an intuitive feel for bids, offers, market and limit orders, and order-price dynamics. The markets are structured as brief (one- or two-minute) "trading days". Users can enter orders, pause the market (to think about things or ask a question). The screen contains a real-time graphic display of the market, which shows the recent history of bids, offers, and trades. Users can replay the market to see the effect of varying their strategies.

In a classroom setting the instructor can specify drift, volatility, length of the trading day, arrival rates of uninformed and informed traders, and many other parameters. Pre-specifying the random seed used to start the simulation ensures that the entire class is facing the same path of the security value. After each market, the user's result is summarized in an encoded string that can be pasted, for example, into an email or Google form. This allows the instructor to collect results. 7EX is not a multiuser exercise, though. The user is always playing against the machine.

7EX was developed in Excel 2010 Visual Basic for Applications on a Windows 7 system. The current release (Version 2) is available for educational (non-commercial) use.

For a quick start, load the spreadsheet (you might have to enable macros). Follow the instructions on the home screen.

Section 2. Using 7EX for self-study

The security being traded can be just about anything. I think of it as a share of stock. Depending on configuration, the market can be calm or volatile. Traders can arrive frequently, or hardly at all. The security has a fundamental value, which changes over time.
A trading “day” is short, just around a minute or two. You can replay the day to see how alternative strategies might have worked.

The players are the user ("you") and the “crowd”. On a floor-based exchange, the crowd consists of people who gather in one place and have an interest or potential interest in trading the security. In electronic markets the gathering is virtual, and the crowd can consist of human traders or automated algorithms. The 7EX crowd contains:

- Patient buyers. The crowd bidders roughly know the value and are willing to buy using limit orders priced a bit lower than their estimate of the value.

- Patient sellers. The crowd sellers enter limit orders. Their offering (asking) prices are a bit higher than what they think the security is worth.

*The patient buyers and sellers are always active. There is always a crowd bid and a crowd ask.*

- Liquidity traders. These traders arrive randomly and buy or sell at the bid or offer. Some of the liquidity traders just want to get their trade done, and they don’t care about the price. They’ll hit any bid or offer. Other liquidity traders are a bit more discerning. If they are buying, they won’t trade unless the asking price is below their (unobservable) reservation price.

- Informed traders. Depending on how you or your instructor has configured the market, some of the incoming traders might be informed. They know the value of the stock at the end of the day. If the offer price is below the closing value, they’ll buy (by lifting the offer); if the bid price is above the closing value, they’ll sell (by hitting the bid).

The 7EX crowd is perfectly capable of trading all by themselves. You can play (simulate) a market without entering any orders yourself. Let’s try it with the demo (demonstration) parameters.

**2.1. Demo**

When you load 7EX, you’ll see a home screen like this:
The home form suggests some market scenarios that you might want to play. The demo market is already highlighted. When you click select market, the demo parameters are loaded, and the play button is enabled.

Clicking play takes you to the trading workstation. Move the forms so that the market form and the chart are both visible:
• The market setup frame has a short description of how the market is configured. By rolling the cursor over the frame, you can get some suggestions about what you might do.

• The control frame has buttons that will allow you to start, stop and pause the market.

• The orders frame has the controls for submitting orders. These are only active when the market is actually running.

• My positions will contain a running score of your positions (shares and cash), profits, and other messages.

• Market will summarize the state of the market: time remaining, current bid and offer, etc.

• Graphics contains a single checkbox that sets the range of the play.

To the right of the market form is a graphic display of recent market developments. When the market is running, this graph will be updated with bids, offers, and trades.

Ready? Hit the start button, let the market run for ten seconds or so, then hit pause.

Here’s what’s happened:

• The crowd started bidding at $19.95 (blue) and asking $20.05 (red).

• Around 5 seconds, there was some sort of negative news. Both the crowd bid and offer dropped. There were several more quote changes in the 5-10 second range.

• At about 7 seconds, a buyer arrived and “lifted” (traded at) the crowd offer. This is marked by an empty circle. Around 9 seconds, a seller arrived (and hit the bid).
Got it? Now click on *continue* and let the crowd trade for the rest of the 30 seconds.

The market is now closed. The bid-ask midpoint is down $0.15 relative to the opening midpoint. By default the graph shows the most recent 20 seconds of activity. By checking *Plot full day* in the *graphics* frame, you can see the full day (here, 30 seconds).

Now click *back to home*. The home screen now has two buttons enabled: *replay* and *replay new*. Both buttons will put you back in the same market scenario.

- *Replay* repeats the same path for the security's value (like in the movie "Groundhog Day").
- *Replay new* takes the same parameters and simulates a new value path.

In general, the evolution of the market depends both on the value path and the incoming trades (including yours). Even with the same value path, if you’re trading, you will change the market. Of course, you can’t replay a real market.

### 2.2. Market orders

When the market is running the *order* frame is enabled, and you can enter orders to buy or sell. The most straightforward orders are "at the market". *Buy market* lifts the prevailing ask price: you buy immediately at the ask. With *Sell market*, you sell immediately by hitting the bid.

Try it. If you have just played the *demo* market, you can click *replay*. (If you’re starting here, click on *select market*, then *play*.)

Now, start the market, click on *buy market*, and then *pause*. 
Your purchase is graphed as a solid black dot at the ask.

The *My positions* frame summarizes the state of your account. You own 100 shares. To pay for these shares, you borrowed $20.05 \times 100 = $2,005. The borrowing is shown as a negative. (Don’t worry, there’s no interest assessed.)

Whenever possible, traders computed positions, profits and losses on the principle of “mark to market” (MTM). 7EX uses the midpoint of the crowd bid and offer as the price mark. This is, at the moment, $20, giving us an MTM value for the shares of $2,000. Net of our cash holdings, we have profits of $-5.00 (a loss).

How did we lose money so quickly? It’s simple. When we bought with a market order, we paid a bit more than the stock was worth, for the privilege of making our purchase exactly when we hit the button. That had a cost.

Now let’s close out the position (sell the shares). *Continue* the market, immediately hit *sell market*, and then *pause*. 
Our shares are at zero. But by buying at the ask and selling at the bid, we’ve paid the full spread.

Market orders, like the ones we just used, are considered active trades. (We’re the instigator.)

2.3. Limit orders

A limit order is a patient order. It is submitted with a price. “Buy limit $19.95” means “Buy, if possible, but don’t pay over $19.95.” Relative to a market order, we are giving up the certainty of an execution, but hoping to buy at a lower price.

“Buy limit $19.95” is an ordinary limit order. If the crowd bid price were $19.90, our price would be more attractive. If the crowd bid changes, though, we might our limit price to change with it. We can do this by attentively watching our screen and revising the price whenever the crowd bid changes. In a volatile market, though, this can require a lot of concentration and fast typing.

To simply order management, many limit orders used nowadays have prices that automatically float with the market. Such orders are said to be “pegged”. In practice, buy limit orders are usually pegged to the market ask price (the opposite-side quote), or the bid-ask midpoint.

7EX allows only limit orders pegged to the same-side crowd quote. These orders are submitted using the up/down arrow buttons.

Let’s try one. Stop this market and on the home screen hit replay. Now as soon as the market starts, enter a pegged limit buy order by clicking once on the up arrow. Then pause the market.
In the orders screen (greyed out because the market is paused), there’s a message that our bid is pegged at $0.01 above the crowd bid. In the market frame, the current best bid is $19.96 ("yours" because it is your pegged order). In the graph, our bid is shown just above the crowd bid.

Now hit continue, run the market for a few more seconds and then pause it again.
• Our pegged bid moves with the crowd bid. When the crowd bid drops, our pegged bid tracks it.

• At around 9 seconds, a seller arrives and hits our bid. Once our order is executed, it is no longer active. (We can immediately resubmit at the last pegged price by clicking on reset bid peg.)

Now try a pegged limit sell. On the price spinners, the up and down arrows indicate pegs that are more or less aggressive. An aggressive pegged bid is above the crowd bid; an aggressive pegged offer is below the crowd offer.

It is pretty clear that if we do all of our trading using market orders, we are incurring spread costs that will be difficult to recoup unless we’re very lucky. We’re always buying high and selling low. With limit orders, though, the spread works to our profit.

Try replaying the demo market using only pegged limit buys and sells. Try to keep your position (number of shares owned) close to zero. Can you make a profit on the day?

Can you hit your own limit orders? Yes. In both 7EX and real-world markets, you can trade against yourself, as both buyer and seller.

A trade with yourself is called a wash trade. Most markets have rules against wash trades. The reason in that many people watch the tape to get a sense of where the market is headed. Trades are normally presumed to be “at arms length”, that is between a competitive buyer and a competitive seller.

A wash sale is, in and of itself, a minor infraction. Sometimes in one firm with many traders, each acting independently, a wash sale might happen simply by chance, without any coordination. A pattern of wash sales, though, particularly if it involves a profitable trading strategy, might be viewed as an attempt to manipulate the market.

Section 3. The markets

7EX is preloaded with about a dozen markets that illustrate different situations. Here is a summary of the markets, and the situations or effects that they illustrate.

<table>
<thead>
<tr>
<th>MarketID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td>Short demonstration: 30 sec, no trend, no informed traders. Good for getting the feel of the simulator and learning how to submit orders.</td>
</tr>
<tr>
<td>DemoLong</td>
<td>Longer version of the demo.</td>
</tr>
<tr>
<td>PriceP</td>
<td>Transient price pressure: 2 min, value doesn't change, no crowd orders. Often when one side of the market is hit, particularly several times in quick succession, orders are withdrawn, and the quote moves away. With the passage of time, if no more orders on that side appear, they revert. Try submitting several buy orders to see the temporary price impact and reversion.</td>
</tr>
<tr>
<td>MarketID</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Uptrend</td>
<td>Trading with an upward trend; 2 min. Here, you know that the stock is tending upwards during the day, so your basic strategy is to buy early. If you buy with market orders, though, the trend will have to be strong enough to overcome the spread and temporary price effects. Try experimenting with limit orders.</td>
</tr>
<tr>
<td>DownTrend</td>
<td>Trading with a downward trend, 2 min. Here, we’re selling short.</td>
</tr>
<tr>
<td>Split1</td>
<td>Temporary price pressure effects make it expensive to trade a lot at one time. In this situation, try to buy 1,000 shares by the end of the market (2 minutes). You can minimize your cost by using limit orders and gradually trading over time.</td>
</tr>
<tr>
<td>Dealer1</td>
<td>The objective here is to make money with limit orders, while staying as “flat” (close to zero-shares) as possible. With these parameters, your profits are generally highest when your limit orders are pegged at the minimum ($0.01)</td>
</tr>
<tr>
<td>Dealer2</td>
<td>Acting like a dealer, 2 min, with price-sensitive liquidity traders. Like Dealer1, except that many of the liquidity traders are price sensitive. They will only trade if they can get price improvement (over the crowd) of at least $0.02.</td>
</tr>
<tr>
<td>InfTrd1</td>
<td>Informed trader; no liquidity traders. When the informed trader arrives, she’s always trades in the direction of making a profit. By watching her buys and sells, you should be able to estimate the closing value.</td>
</tr>
<tr>
<td>InfTrd2</td>
<td>Informed and liquidity traders.</td>
</tr>
<tr>
<td>RptTest</td>
<td>Test of return code encryption. The result code (generated by the <em>report results</em> button) is normally an unencrypted string containing user and market identifications and a summary of results. In RptTest, the string is encrypted. See section 4.4.</td>
</tr>
</tbody>
</table>

### Section 4. Using 7EX in the classroom.

#### 4.1. Overview

7EX is suitable for independent out-of-class exploration or for coordinated in-class exercises. It can be used as-is, as long as the set of predefined markets meets the instructor’s needs. The instructor can also define new markets. This involves specifying new sets of parameters. This is explained more fully in section Error! Reference source not found.

7EX also provides a basic functionality that helps the instructor keep track of users’ results. After a market is stopped or closed, 7EX summarizes the outcome (trades, profits, etc.) together with user identification in one return code (a long string). The user then pastes the string into an email (to the instructor) or into a web form (set up by the instructor). The Google Forms app is easy to use for this purpose (and is currently free). The process is described more fully in section 0.
4.2. The market parameters

The parameters worksheet contains a table of parameters that can be altered or expanded. Rolling the cursor over a column label displays a brief explanation of what the parameter does and its default value.

All events are modeled as Poisson arrival processes, with intensities that are specified as “expected events per minute.” Public information events, for example, are governed by \( m_{\text{Intensity}} \). This defaults to 10, i.e., on average about one event every six seconds. Other intensities govern the arrival of liquidity traders and informed traders.

The program is structured as if each class of trader only contains one individual. This is a programming convenience. Poisson intensities are additive. The arrival process for traders has the same properties if we’re simulating 100 traders with intensities of one arrival per minute or one trader with intensity 100 per minute.

The parameter \( \text{timeLimit} \) defines the duration of the market (in seconds).

The underlying value process is denoted \( m_t \). It evolves as a jump process. Let \( \tau_k \) denote the \( k \)th (simulated) public information event time. Then

\[
m_{\text{tau}_k^+} = m_{\text{tau}_k} + u_k
\]

The size of the jump is distributed normally: \( u_k \sim N(\mu_k, \sigma_k) \) where:

\[
\mu_k = m_{\text{Drift Day}} \frac{\tau_k - \tau_{k-1}}{\text{timeLimit}} \quad \text{and} \quad \sigma_k = m_{\text{Sd Day}} \frac{\sqrt{\tau_k - \tau_{k-1}}}{\text{timeLimit}}
\]

Price pressure effects are modeled separately on the bid and offer side.

as exponentially weighted averages, updated whenever there is an event, or when one second has elapsed. Denote the current time as \( \tau \), and \( \tau_{\text{lastUpdate}} \) as the time when the average was last updated.

The price pressure term is defined on the offer side as as \( \text{gamma} \times MA_{\text{Buy}} \). \( MA_{\text{Buy}} \) is exponentially weighted moving average of prior buys. It is updated whenever there is an event, or when one second has elapsed. Denote the current time as \( \tau \), and \( \tau_{\text{lastUpdate}} \) as the time when the average was last updated. Then the update rule is:

\[
MA_{\tau}^{\text{Buy}} = w \times MA_{\tau_{\text{lastUpdate}}}^{\text{Buy}} + \begin{cases} 1, & \text{if there is a purchase at time } \tau \\ 0, & \text{otherwise} \end{cases}
\]

The weight is

\[
w = \exp[-\lambda \times (\tau - \tau_{\text{lastUpdate}})]
\]

\( MA_{\text{Sell}} \) is defined similarly.

The crowd best offer is
\[ CBO_t = \text{RoundUp}[m_t + \lambda \times MA_{Buy}^t] \]

where \text{RoundUp} rounds up to the next higher tick (penny).

Ordinary liquidity traders, governed by arrival intensity \( tLiquidityIntensity \), buy or sell with equal probability. Price-sensitive liquidity traders \( (tLiqPrcSensIntensity) \) will only buy or sell if they can get price improvement (relative to the crowd bid or offer) of at least $0.02. That is, they will only trade if the user has a pegged limit order that betters the crowd by at least $0.02.

4.3. Configuring new markets

On the parameters worksheet, the coordinator can define new markets by either overwriting existing markets, or by adding new rows at the bottom of the parameter table. If the size of the table has changed, the named range \text{parmTable} must be adjusted. In the Excel menu, click on Formulas, and then Name Manager. Adjust \text{parmTable} so that it fits your new table.

If you don’t want the users to see your market parameters, hide the parameters worksheet, and in the Permissions menu (off of Excel’s File) menu, Protect workbook structure (with a password). Then post the spreadsheet to your own web site.

4.4. Keeping track of users’ results

The report results button generates a return string that summarizes the outcome of the play. The first time it is clicked, the user is prompted for an ID string. It should be something that the coordinator can use to identify users (like an email address).

The return string is a comma-separated value string that includes the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Userid</td>
<td>Whatever the user entered when first prompted.</td>
</tr>
<tr>
<td>marketID</td>
<td></td>
</tr>
<tr>
<td>nPlay</td>
<td>The number of consecutive times this marketID has been played.</td>
</tr>
<tr>
<td>mRandSeed</td>
<td>The random seed that was used to initialize the simulation draws.</td>
</tr>
<tr>
<td>ranToClose</td>
<td>“true” or “false”: did the user run the market all the way to the close?</td>
</tr>
<tr>
<td>myShares</td>
<td>Final share position</td>
</tr>
<tr>
<td>myCash</td>
<td>Final cash position</td>
</tr>
<tr>
<td>myMMValue</td>
<td>Final mark-to-market value of the shares</td>
</tr>
<tr>
<td>myProfits</td>
<td>myCash+myMMValue</td>
</tr>
<tr>
<td>myBuys</td>
<td>Number of buys (both active and passive)</td>
</tr>
<tr>
<td>mySells</td>
<td>Number of sells (both active and passive)</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>myActive</td>
<td>Number of active trades</td>
</tr>
<tr>
<td>myPassive</td>
<td>Number of passive trades</td>
</tr>
<tr>
<td>MyWash</td>
<td>Number of wash trades.</td>
</tr>
</tbody>
</table>

The return string is copied to cell A1 on the home worksheet, and is also copied to the clipboard. By default, it is not encrypted.

Once the return strings are collected in one column in a spreadsheet, they can be converted using the Excel text-to-data feature.

There are two ways of collecting the return strings. Each student can self-report the return code by email, and the instructor can cut and paste from the emails. This is feasible, but cumbersome.

An alternative is to use the Google Forms app. Set up a Google account, and create a new form that has one field (a text field, with some clever name like "result"). Embed a link to the form on the same page where you posted the spreadsheet. Instruct the users to click on the link, which will open a browser window on the form. Then instruct them to report their results by pasting in the return code and clicking on submit. The responses will be collected in a Google spreadsheet that you can download and analyze.