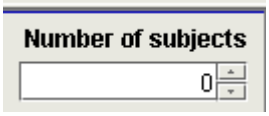
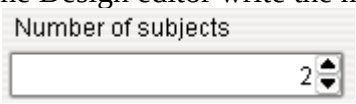
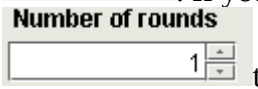
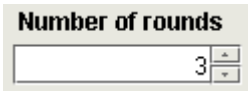
1. Double click on 

that is located on your desktop and the editor like the one below will appear.
2. In order to create a free form experiments you will need the following basic building blocks:

a. The first thing is to specify the number of subjects who will be playing against each other in one session during the experiment. Under 

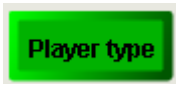
that is located on the right side of the Design editor write the number of subjects that will participate in the session : 

. If you want subjects to repeat experiment several times write under 

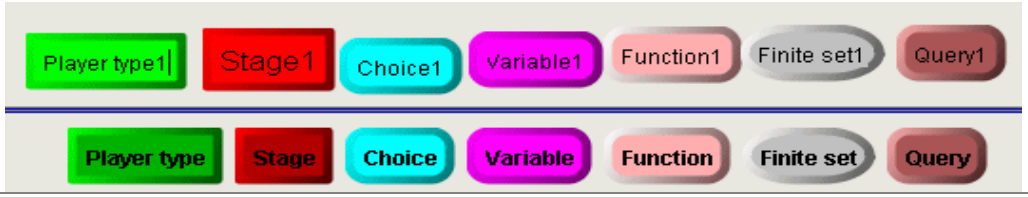
the appropriate number. If you write 

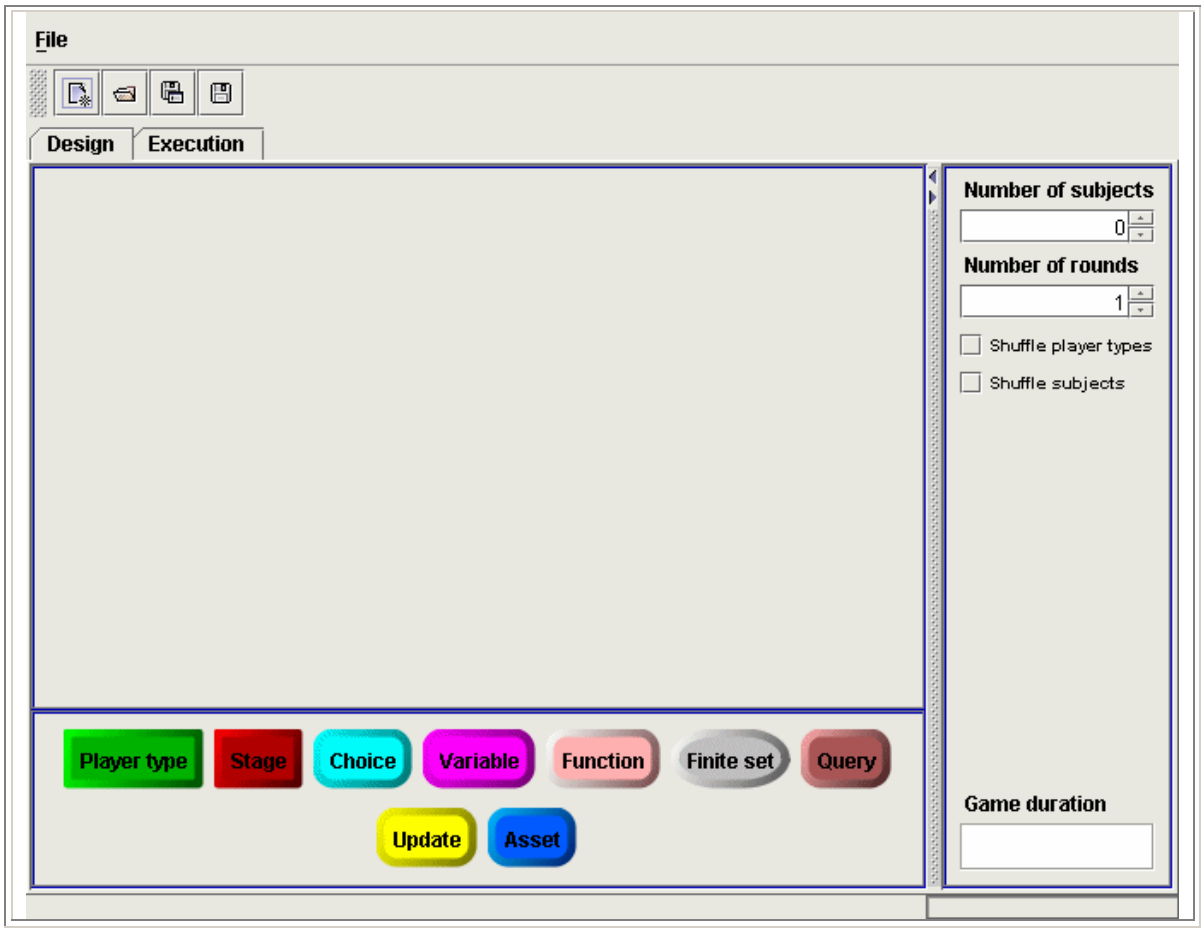
the experiment will be repeated three times.

b. The elements that we need to construct the can be created by dragging and dropping the element at the bottom of the design editor. For example 

was created by dragging and dropping 

into the design window and similarly on all the other icons.





I. ComLabGames Tools

a. Design Window
b. Function Editor Assistant
c. Execution
d.

III. Expressions

a. ComLabGames provides mathematical *expressions*. Each expression is composed of: 'operand operator operand' (i.e. expression can consist only of an operand). For example, expression '1+2' means, that we have operand '1', operator '+' and operand '2'. The building blocks for the expressions are

- Operators

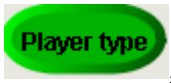
and operands:

- Functions
- Numbers
- Variables
- Choices

b. Function Editor Assistant

Before we explain each of the building blocks for any expression we would like to explain ***Function Editor Assistant***. It consists of all the operators and operands. At the command prompt for 

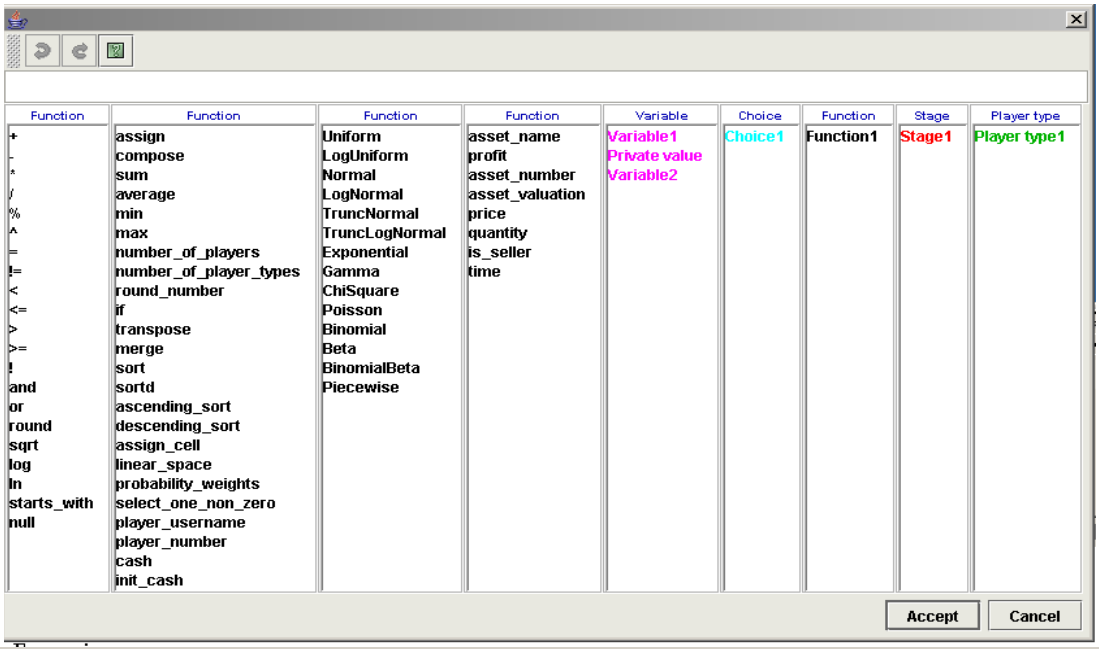
, 

, 

, 

, 

(double click on each of these elements you will get a command prompt), in the command prompt for output stage and output variables, click on "**ALT"** and "**double click"** at the same time and ***Function Editor Assistant*** will appear**.** You can write the expression directly into the command prompt for this Editor and when you finish click on "Apply" and the program will automatically update this expression into the element from where you started the ***Function Editor Assistant.***  is a help button for the editor,  is Undo button and  is Redo button. To write an expression in the ***Function Editor Assistant*** double click on any function, variable, choice, stage or player type or you can write directly into the editor. All variables should be violet color. If you write directly, click on **"Crtl" "Shift" "s"** and start writing the variable name in violet color. **"Crtl" "Shift" "c"** will change to turquoise color and **"Crtl" "Shift" "n"** will return the mode back to the black color.



a. Operators

Expressions use familiar arithmetic operators and precedence rules.

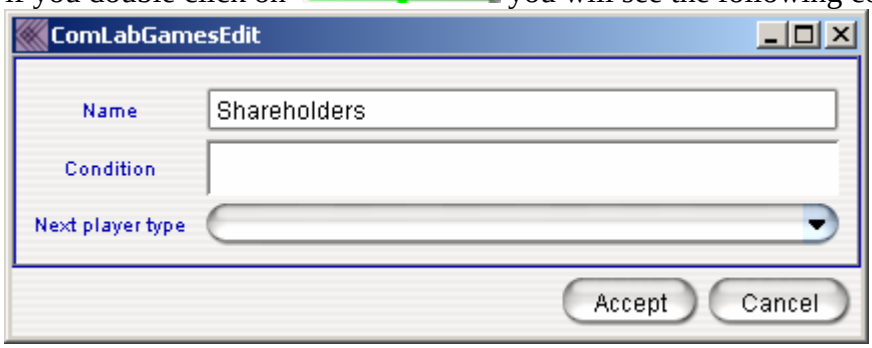
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	
^	Power

= != < <= > >= and or	Equal Not equal Less Less or Equal Greater Greater or Equal

All these operators are listed in the first column up to 'or' of the ***Function Editor Assistant.*** Each operator can perform a certain type of data. For example '+' works with numbers and text, but not with Boolean. However 'or' work only as a Boolean (operand can be text but the possible values if the text is "true" and "false").

Player type definition

1. Once you drag and drop the player it will have a default name Player type1.
2. You can leave this name or change it by clicking on Player type1 icon and change it. We selected the name Shareholders.
3. Note if you double click on Shareholders you will see the following content:

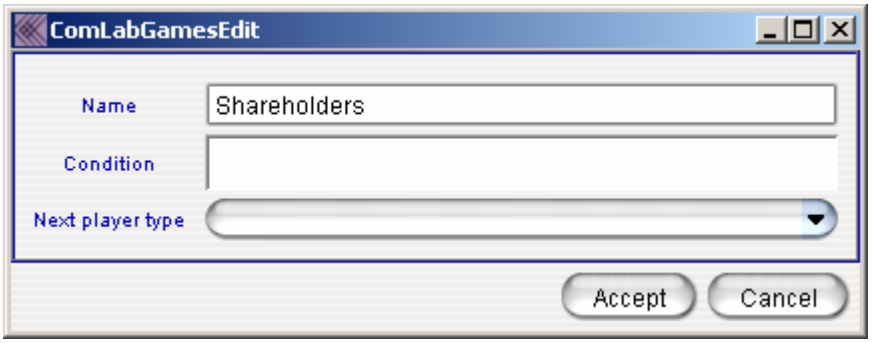


The purpose of this window is for naming player types, in the **Name** command box if you have not named it before directly on Player type icon , and determining how experimental subjects are assigned to player types. The **Condition** command box

determines whether the logon identity of a subject meets the specified conditions for being that type of player. The **Next Player** command box sequentially determines the order in which the conditions for the various player types are checked.

a. Naming player types

If you have not already renamed directly Player type on the created icon, the top row in the **Player** editor is allows for naming the player type. In this example we renamed Player type to Shareholders by typing "Shareholders" in the space following **Name** and then clicking **Accept**.



b. How the conditions are applied

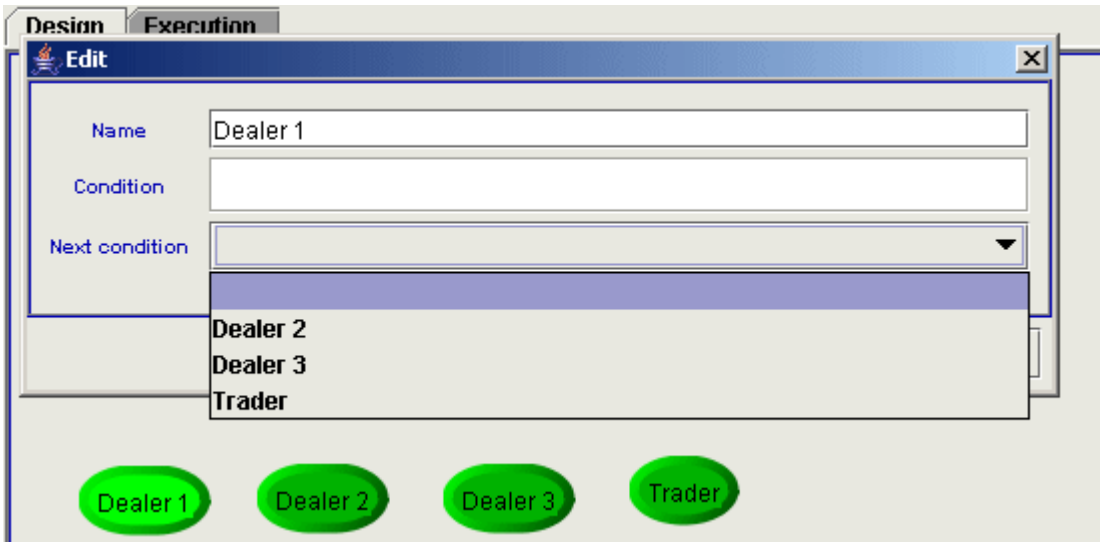
If there is more than one player type, then the command line for **Condition** must be completed for all but one of the types. We explain conditions are defined in the next subsection. Under the name of each player type

The second row of this editor, **Condition,** determines how subjects, defined by their log names, are assigned to player types. Write which conditions a subject must satisfy to be assigned to that player type in the command line for **Condition**. You can use the **Function editor assistant** to access all the available **ComLabGames** functions and elements created already by using the command **"ALT"** and **"double click"** at the same time.

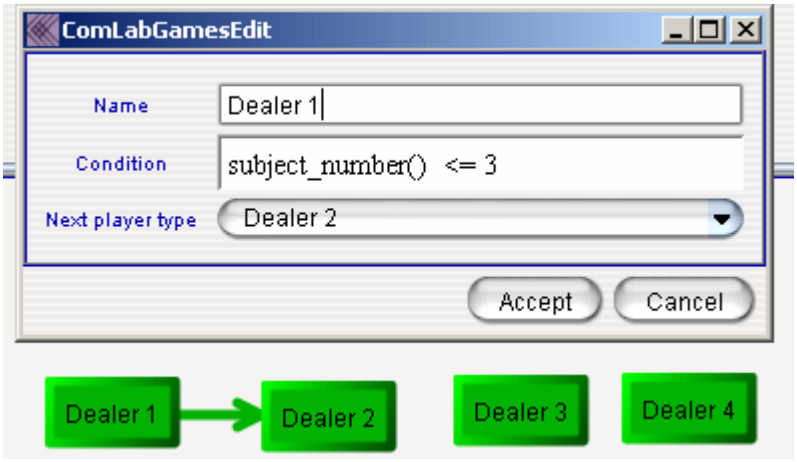
If a subject does not satisfy the requirements written in the **Condition**, and the **Next Player** command line is not empty, he is assigned another player type. In that case the subject log name is checked against the condition of the **Next Player**. In this way the order in which conditions are applied precisely determines how subjects are assigned to player types. When only one player type is created then **Next Player** is left empty.

When more than one player type is created, they are all appended to the **Next Player** box automatically and the list can be viewed by clicking on . The example below shows four player types being created, namely Dealer 1, Dealer 2, Dealer 3 and

Trader.



If you create the four player types named above, and select Dealer 2 after clicking on ▼ in **Next Player** of the Dealer 1 editor, then subjects who log on to the experiment and fail to meet the criteria for Dealer 1, are then checked against the **Condition** for Dealer 2. In fact the program draws an arrow from Dealer 1 to Dealer 2, shown in the figure below.



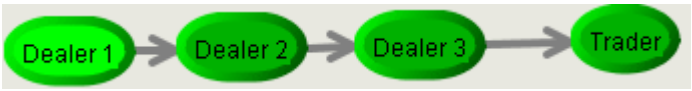
c. Assigning subjects to player types

The path generated by the arrows from successively selecting **Player** icons from **Next Player** commands filter Player types are disjoint and therefore it does not matter with which player type you start and the order of the Next condition. You can look at this like a filter. Player type is calculated in predetermined order and the first condition

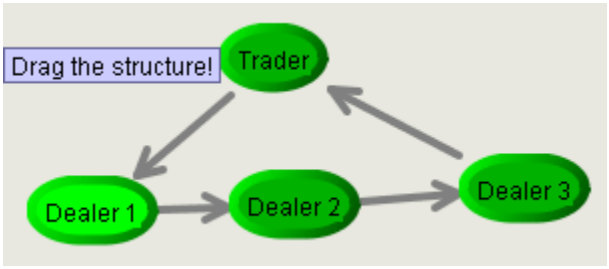
that succeeds (for which the Boolean expression returns the result "true") is assigned to a subject. For each subject the calculation of each player type starts from the beginning. The marginal conditions are executed in the following way:

empty condition is equivalent to "true"

the last Player Type in the chain always succeeds independent of the condition.



If we have a loop between the player types like in the example below, then in the worst case (all the conditions are false) subjects will get assigned to the player type that is the first in the second time of evaluations. In our example this is "Dealer 1"

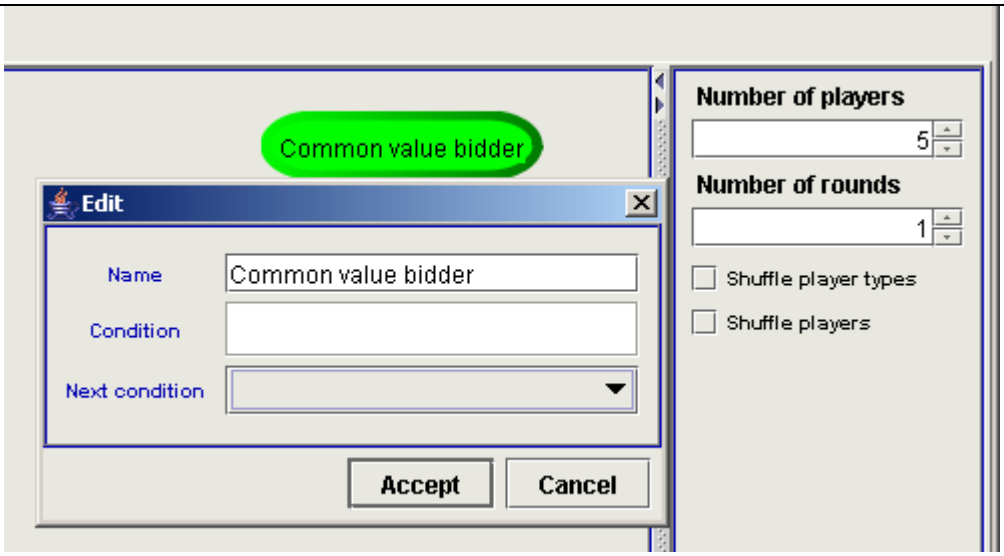


d. Examples of Expressions for determining Player Types

- **Only ONE player type is created for the experiment.**

We do not assign "Condition" and "Next condition".

In this case the empty condition is "true" which means that each subject who logons to the game will be assigned to a player type (i.e. common value bidder). In our example there are 5 players specified under "Number of players" which means that there will be 5 common player types assigned to one session and the next 5 players to the second session (i.e. with this we determine how many players play in one group (session).



MORE than ONE Player Type is created for the experiment the rules can be the following:

1. *Condition*: Dealer 1: subject_number() <= 3
Dealer 2: subject_number() <= 7
Dealer 3: subject_number() <= 15
Trader: (empty condition)

Note: subject_number() starts with number 1.

In this case the first three subjects who logon will be assigned Dealer 1, the forth to seventh subject will be assigned to Dealer 2, the eighth subjects to fifteenth subject who logon will be assigned Dealer 3 and the rest will be assigned to Trader. We should be careful that we have more than 15 subjects defined under

for each session.

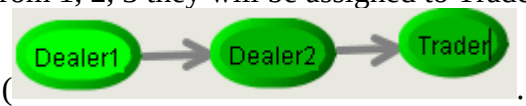
2. *Condition:* Dealer 1: subject_number()%2=0
Trader: (empty condition) or subject_number()%2=1

In this case the second player to login, 4th, 6th,.. will be assigned to Dealer 1. The others will be assigned to Trader. Note that operation with "%" results in the residual from division (subject_number()%2=0 means that number even numbers will have residual zero).

3. *Condition:* Dealer1: starts_with(subject_username(), "1")
Dealer2: starts_with(subject_username(), "2")
Trader: (empty condition)

The function start_with (,) allows to allocate subjects according to the first letter/number in their username. In this case subjects whose username starts with number 1 will be assigned Dealer 1, subjects whose user name starts with 2 will become Dealer 2 and the rest will be traders. If we have letter instead of numbers, note that they are case sensitive.

4. *Condition:* Dealer1: starts_with(subject_username(), "1")
Dealer2: starts_with(subject_username(), "2")
Trader: starts_with(subject_username(), "3")

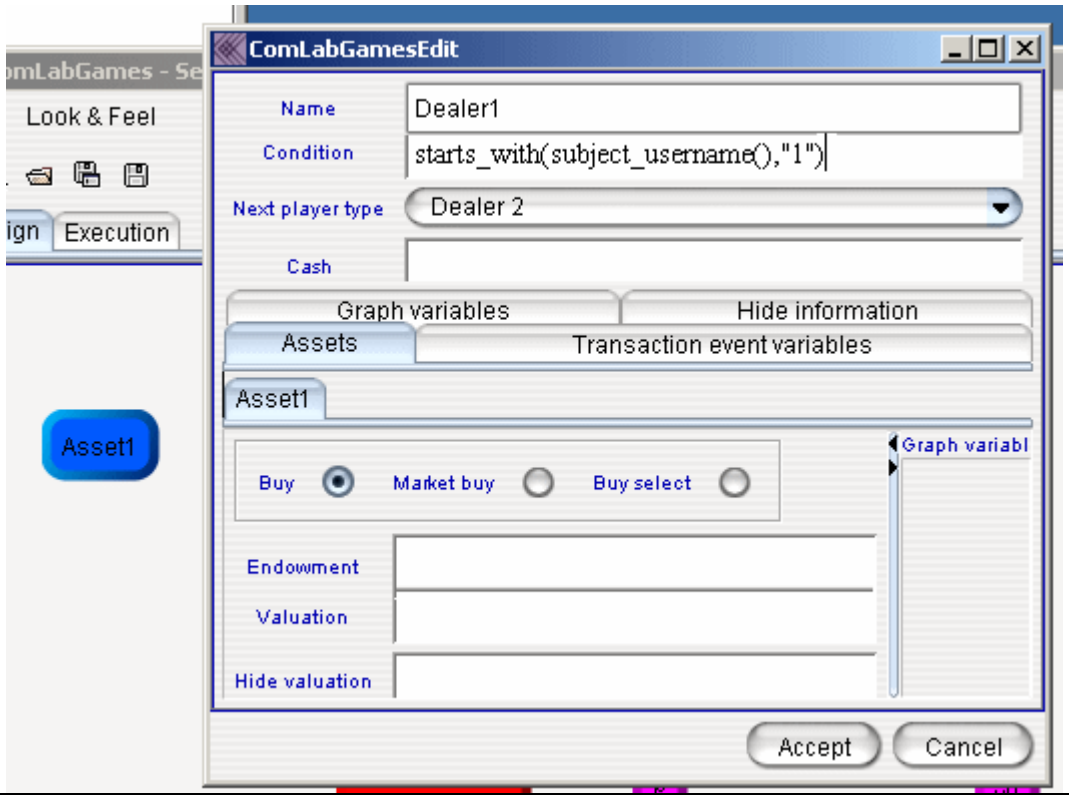
If Traders are given the condition, in the example above, subjects whose user name starts with 3 will be assigned to Trader. However if somebody starts username with anything else apart from 1, 2, 3 they will be assigned to Trader if there is no loop in the "Next condition" (

.

If there is a loop then they would assigned to the player type where the loop start the repeat, in our example with Dealer1:

e. Player type window when an asset is added

- If an asset is created than the conditions related to market is added to Player Type window. The details of these characteristics will be discussed under **Market design.** See below how the Player Type window is automatically updated when an asset is defined and three new pages are added: "Asset", "Transaction event variables", "Graph variables" and "Hide information"

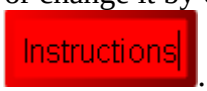


characteristics

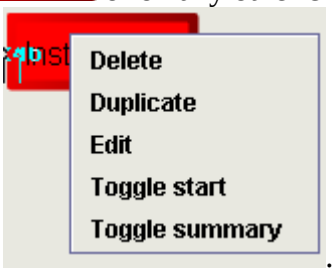
1. Drag 

from below the line in the design window and drop it into the design part of the window 

.
2. You can leave this name or change it by clicking on 

icon and change it. We selected the name 

.
3. Right click on 

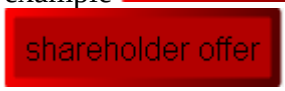
or on any other created stage and the program lists the following options: 

.

a. "Delete" option delete the stage
b. "Duplicate" option exactly duplicates the content of the stage with all the variable updates and state transitions. We will talk about this later
c. Edit allows you to go to the stage editor. Double clicking on the 

will do the same job.
d. Select the Toggle start if you want the particular stage to be the starting stage from where the experiment starts. **Note you HAVE to define a STARTING stage.** By default the first stage that you created is a starting stage. The starting stage has a lighter red color. In our case 

is a starting stage and for example 

is a stage that is not a starting stage. If you want 

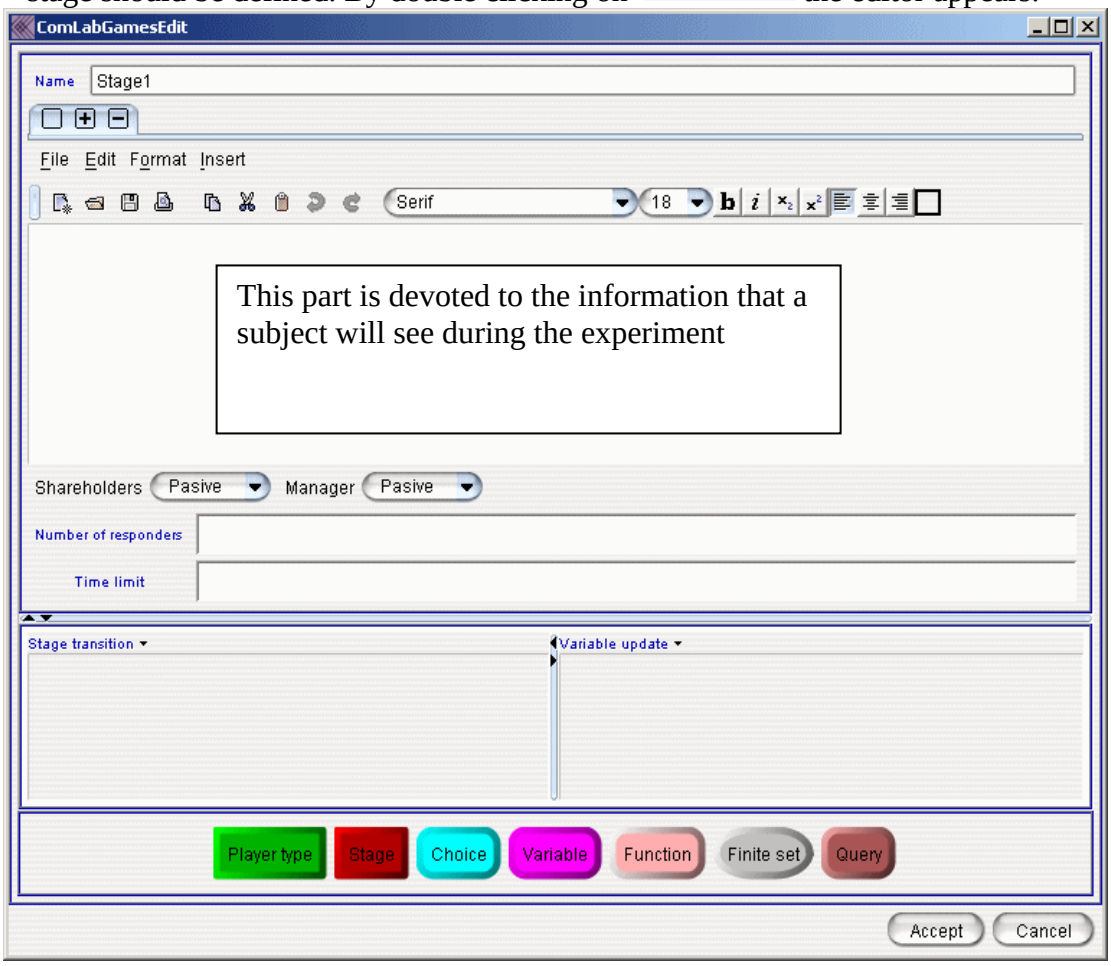
to become a starting stage select, Toggle start and it automatically changes the color 

to indicate that the games starts with this stage.
e. Select Toggle summary to designate the stage to summarize results. You do not need to have State transition pointing to the summary stage if you select this

option. The program automatically ends in the Summary stage if this option is selected. At the end of the experiment we usually like to show the summary result and we can designate the stage to this although it is not necessary. If you select this option the summary stage letters are yellow: Results. Note, if you select more than one round, Results will appear only after the last round.

Stage functions

1. Double click on created stage to open a stage editor where all the content of the stage should be defined. By double clicking on Stage1 the editor appears:

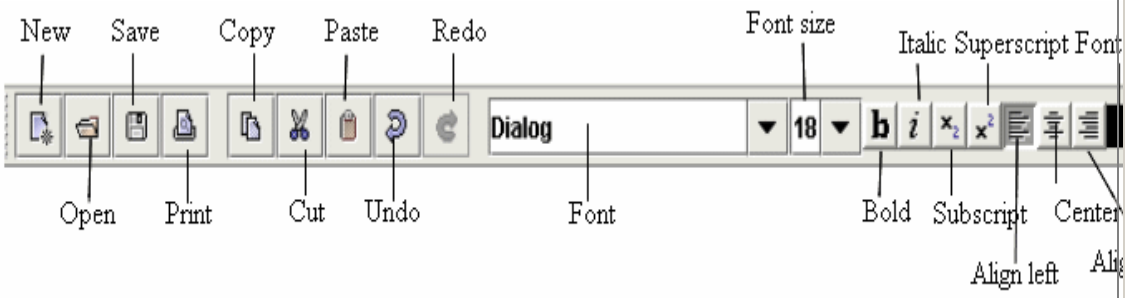


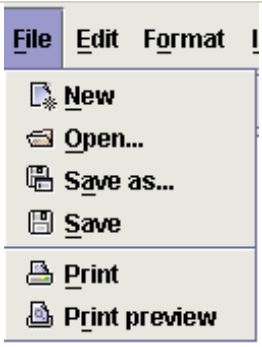
2.  Each stage can have several pages within a stage. With this option we can create simultaneous move games where a page is assigned to a specific player type to make a decision.

a. Click on  to add a new page:  . The page that is active is highlighted in blue.

b. Click on  to delete a particular stage. If we had two stages to begin with  and if we clicked on  on the highlighted stage this stage will be deleted.

3. The toolbar buttons are identical in function to many of the menu commands. Briefly point the mouse at each button to display a *tooltip* identifying the button and its function. The standard toolbar contains buttons for the most frequently used commands on the File and Edit menus.





To open, save, print an existing stage, go to File menu in the stage

Format menu includes bold, italic, subscript, superscript, strike through, left alignment, center alignment, right

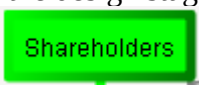
Edit menu includes Copy (Ctrl-c), Cut (Ctrl-x), Paste (Ctrl-v), Undo

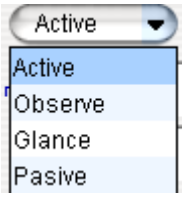
Insert menu is empty when no variables or choices are created. If a new variable or a choice is created it is automatically shown in the Insert menu like in the example below.

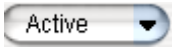
Variables are in violet color and choices are in turquoise color

editor	alignment, foreground (font color), background color, insert images and insert character.	(Ctrl-z) and Redo (Ctrl-y)	

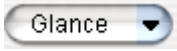
4. 

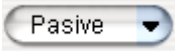
   a. The player types that were created on the design stage will automatically appear in every stage. In our example  and  were the two player types created and in each stage we have four options to define what exactly each of the player types should do on that stage/page. These options are: Active, Observe, Glance and Passive.



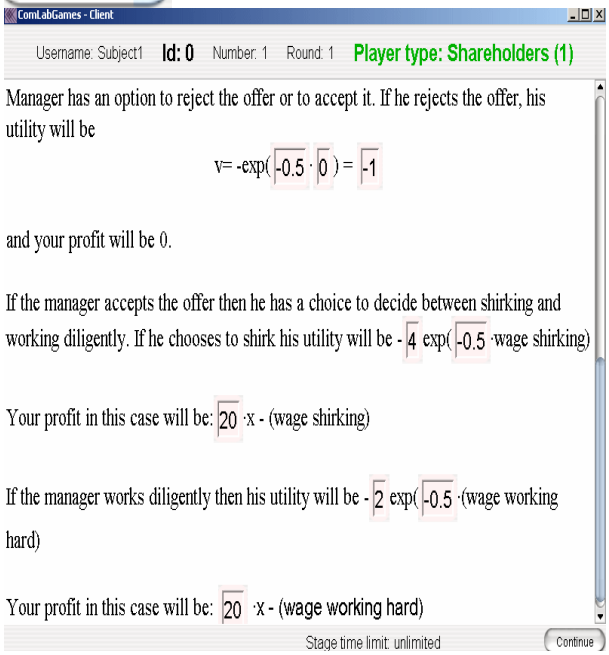
   b.  means that a player type who was selected to be active makes decision in that stage.

   c.  means that a player type sees exactly the same information as the player who is making a decision except that he is informed that he/she cannot make a choice. This player cannot press the “Continue” button.

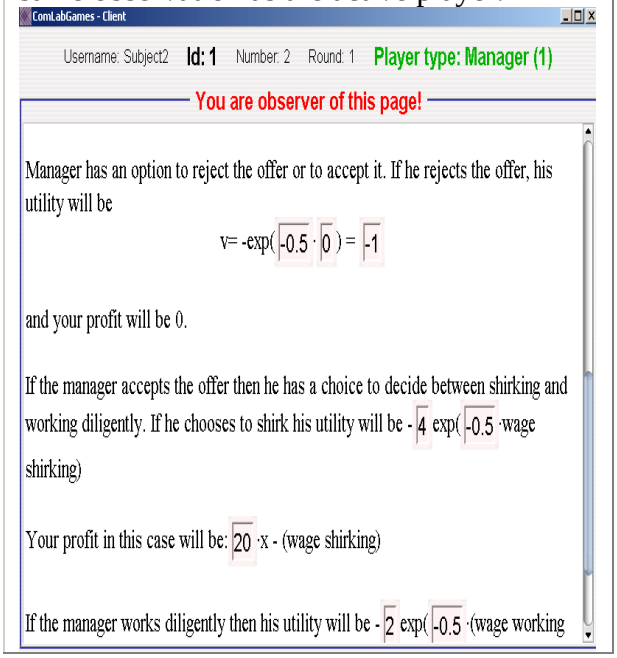
   d.  means that a player type sees everything that an active player type does except he does not see the variable values.

   e.  means that a player type cannot see any information displayed on the page. NOTE, if all player types are “Passive” then the program skips that stage automatically. This option can be used when some variables have to be updated before subjects observe them on the next stage.

Active

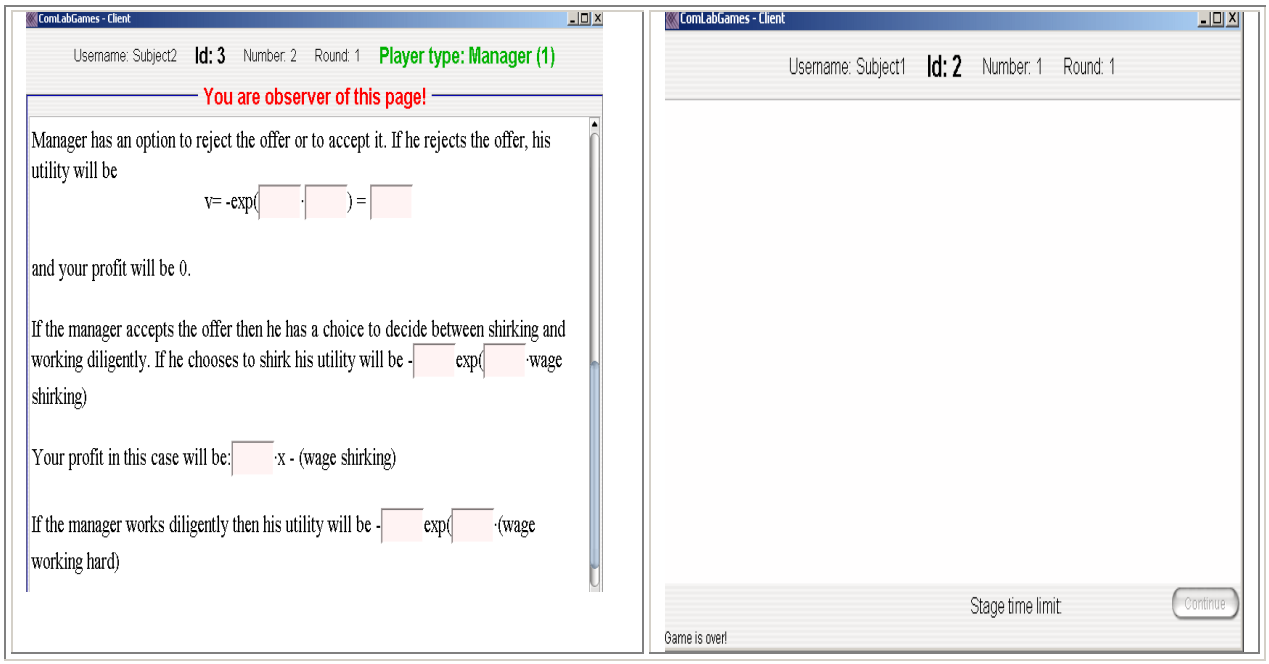


Observe Subjects observe exactly the same observation as the active player.



Glance Subject does not see any variable values

Pasive



4. Number of responders

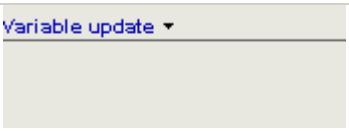
Here you can define number of responders that need to make a decision before you move to the next stage. By default it means the number you specified on the design stage: Number of subjects 1. When you are designing the individual decision making experiment this part is irrelevant.

5. Time limit

Writing a time limit if you want to limit the decision time for a stage. If you write for example Time limit 60 it will mean that a subject has 60 seconds to make a decision. If he/she does not make a decision in 60 second the program will move to the next state defined in the transition update.

6. Stage transition

At the bottom of the stage you will find Stage transition that will allow you to define the path from one stage to the next stage. The function of the stage transition will be defined in details under Stage transition.



At the bottom of the stage you will also find Variable update that will allow you to define the variable to be updated.

Entering and entering text that a subject will see during the experiment

a. The example below is a part of the text written in the 

stage.

Instructions

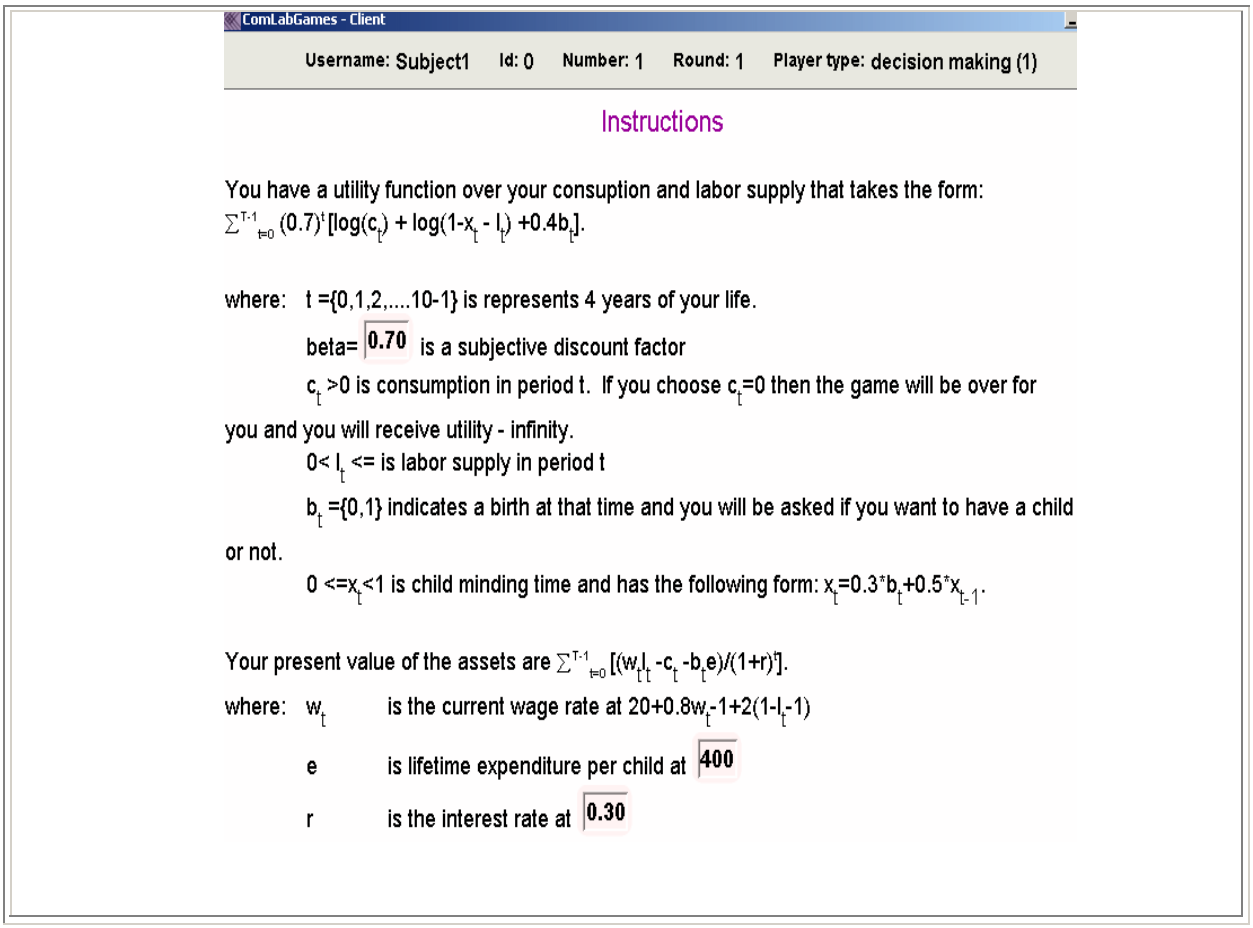
You have a utility function over your consuption and labor supply that takes the form:
$\sum^{T-1}_{t=0} (0.7)^t [\log(c_t) + \log(1-x_t - l_t) + 0.4b_t]$.

where: t ={0,1,2,....10-1} is represents 4 years of your life.

beta= 

is a subjective discount factor

c. When the game is player a subject will see just text part of the stage and the instructions that are shown above will look like this to a subject:



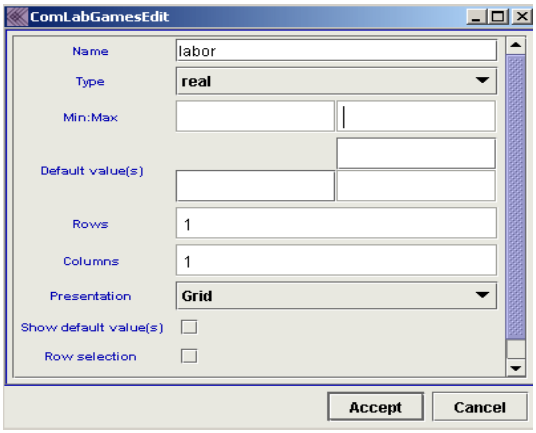
Choice functions

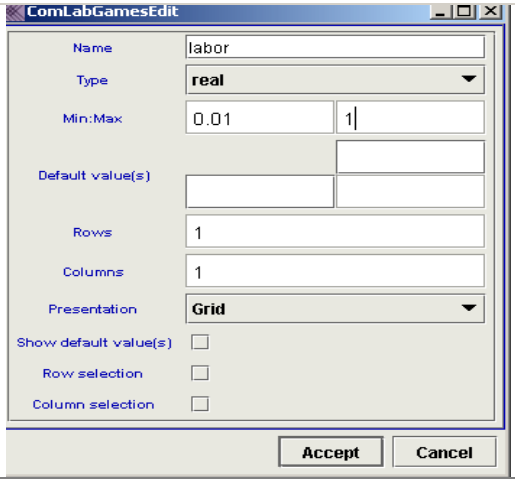
During the experiment subjects will make choices and therefore we have to create a type of a Choice for a subject to write or select. We will first define choice functions and then list several options and appropriately determine how those choices were defined and how they were inserted into a stage.

1. labor choice was created by double clicking on the Choice at the bottom of the design window or by double clicking on the Choice icon at the bottom of the stage. The choice was renamed to labor.

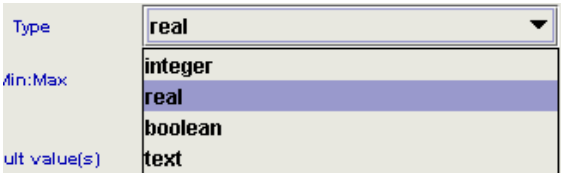
a. The choice editor for labor looks like

this:





The original type of a choice can be:



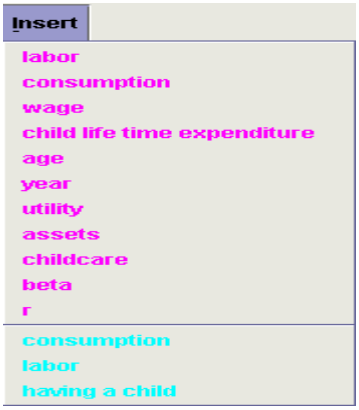
For our choice 

we selected real.

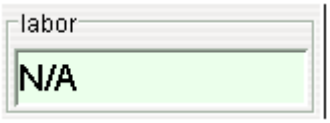
We wanted that subject makes the decision in the range between 0.01 and 1. Therefore we specified Min: Max to be:

b. These are the only options that we selected for labor choice. Once the choice is selected click on Accept button at the bottom of the Choice editor.

d. Insert the specified choice into the appropriate stage using the insert function in the stage:



e. Once it is inserted into the stage it will appear like this in the moderator's stage editor:

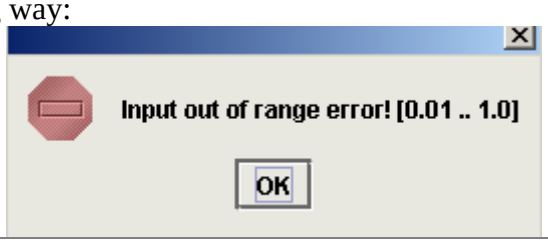


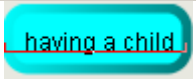
Given that we have not specified a default value it appears as N/A.

f. A subject will see 

choice in the following way:

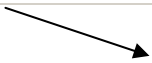
Subject is expected to enter a number between 0 and 1. Given that we specified the minimum and maximum values subject will not be able to enter any number that is lower/higher than the specified minimum/maximum and the program will prompt a subject in the following way:

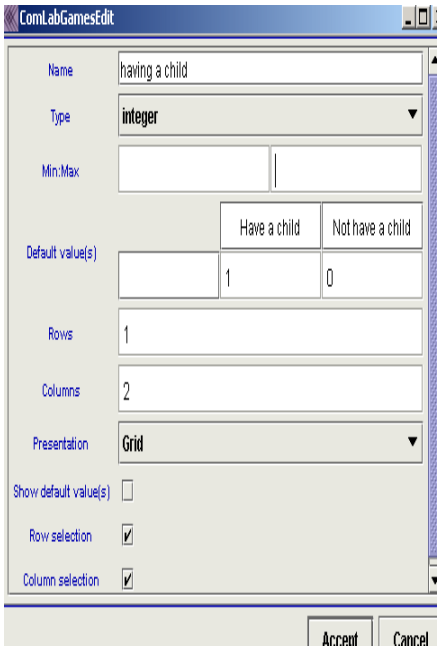


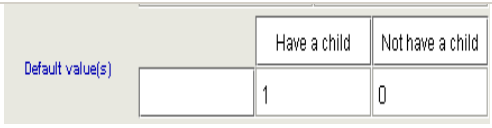


choice has following characteristics:

	a. Integer was specified for the type b. The default numbers for this case were set at 1 for Have a child and 0 for not have a child. We labeled each column.







The row is not labeled. Recall that for the 

choice we left it empty.

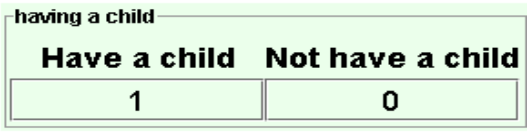
c. In order to create two possible selections: 2 columns and 1 row were specified.

d.

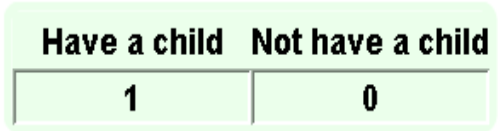
In order for subjects to select ONE cell at the time select both Row selection and column selection.

If you want subjects to select just rows  for Row selection and nothing for Column selection. Similarly if you want subjects to select between columns.

e. Insert the choice in the stage:

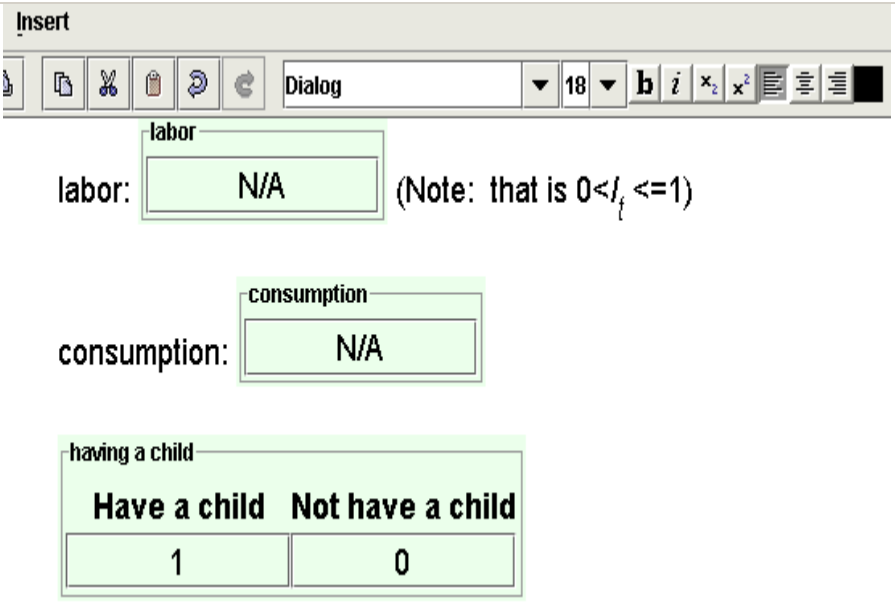


e. Subject's viewing of this choice looks like in the example below



If a subject selects: Have a child, this cell is highlighted in green.

Examples of several choices that were inserted into one of the stages



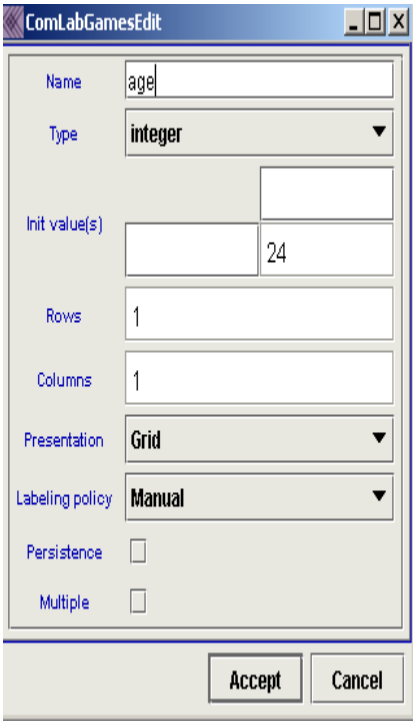
Variable functions

Variable has to be first defined in order to use its value in the stage for subjects to see the value of the variable, to Variable update and to use it in the Stage transition .

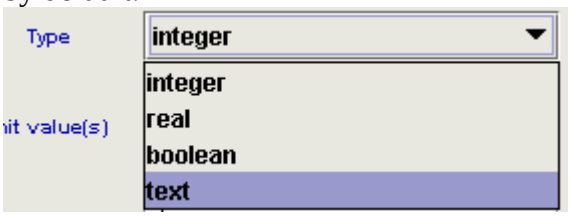
1. Creating Variable with initial values that we want to display in the stage before they are even updated. Below are some examples that subjects see during the experiment with their initial value specified. You might want to tell them what their age is, year, initial wage, interest rate that we called r, discounted factor that we called beta, child expenditure…

a. The variable editor for 

looks like this:



1. Variable type can have the following options by default.

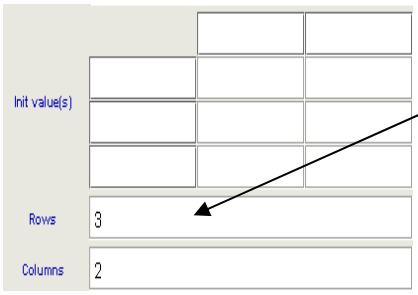


In our example we selected integer.

2. The initial value can be a number like in the example

- it can be a draw from a distribution like the following example:

where a number will be drawn from the uniform distribution with minimum 10 and maximum 20. By clicking “Alt” ”double click” you can access “Function Editor Assistant” (see the section on this)
- We can insert a function that we create as well.



3. By default there is 1 row and 1 column.

can have several rows and columns. Just type the appropriate number:

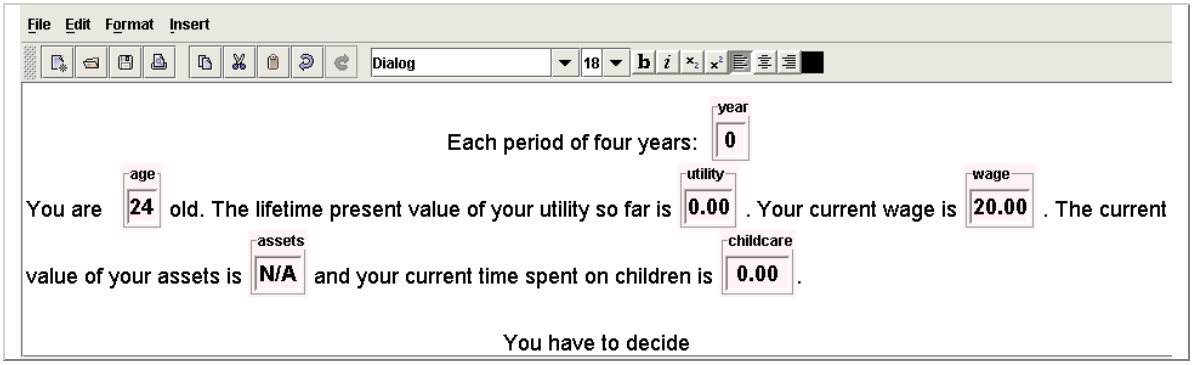
If are set to zero or just row or a column then program will find the right dimension depending how the updated variable is specified.

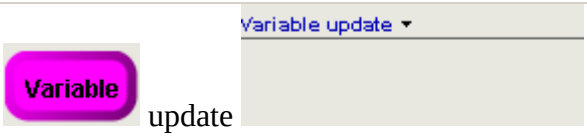
Select persistence if the same values for the 

should

be carried from one round to the next.	 needs to be selected when we have more than one subject per session and we want each subject to get a different draw from the distribution or when variable is updated the appropriate value will be linked with the right person. You do not need to worry which subject made what decision. Multiple variable will do this for you. Note: for individual choice decision we do not need this option.

Inserting  into the stage so subjects can see the values of the variables:
In the example below we inserted year, age, utility, wage, assets and childcare variables. As you see some of them have initial values and for some of them the value will be determined only with updating which is the next section that we will cover.

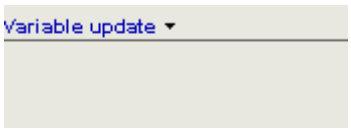




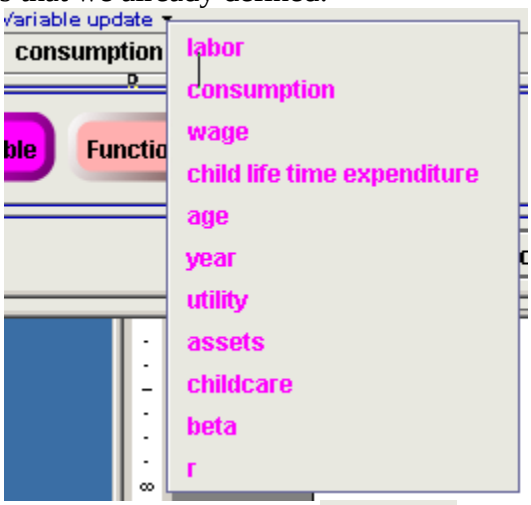
update

Each variable can be updated from the current stage to the next stage. The following condition have to be satisfied:

a. Variable has to be defined and you can verify it by going to

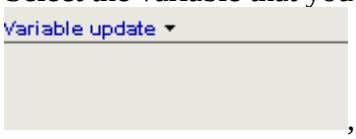


and clicking on the arrow. In our case we have the following variables that we already defined:

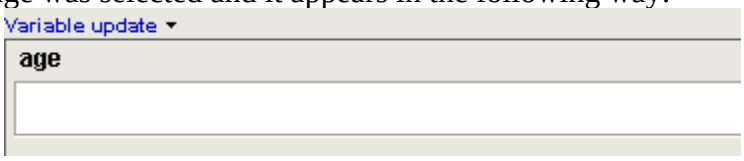


If a variable is not defined just double click on icon at the bottom and specify the parameters in the variable editor.

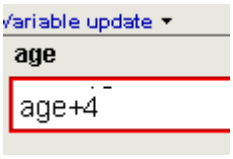
b. Select the variable that you want to update by going to the

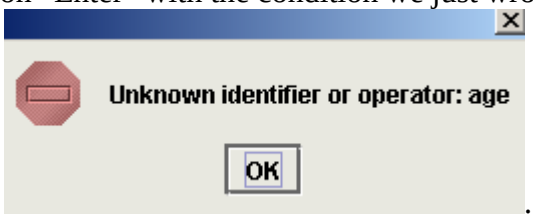


, clicking on the arrow and selecting the variable. Variable age was selected and it appears in the following way:



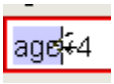
c. Now we want to write how this variable should be updated. In our case we will write the following:



. Note that a red rectangle appears once you start editing a variable editor. If you click on “Enter” it will disappear if the condition is written correctly. Let’s us try clicking on “Enter” with the condition we just wrote above. The program returns the message: 

.

Note that age is a variable and it has to be in pink color. There are two options how to color the variable. The first one is to go with the mouse on the age and select it:



and then at the same time click and hold “Ctrl” “Shift” “v” and the age will be colored in pink.

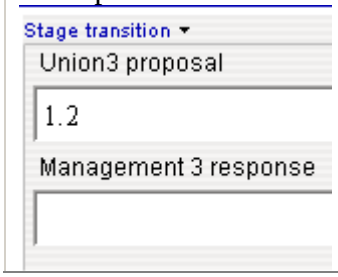
Stage transition

Basic concepts to remember of how the program calculates the transitions

1. First all the expressions are calculated and an empty space is also treated as an expression.

2. Then the program expects if all expressions are either Integer, Real or empty. If this is the case then each empty space receives the weight 1. The sum of all the expressions represents the total weight. Depending of the results a certain stage is selected.

Example:

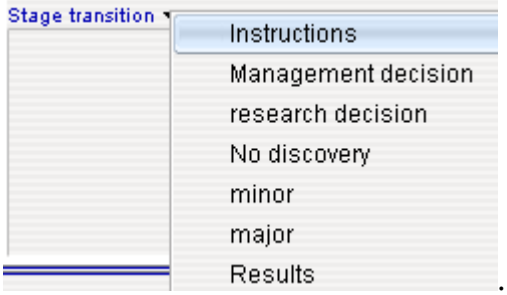


If the first expression for the transition to one of the stages has a weight 1.2 (in the example on the left 1.2 is the weight for transition to “Union3 proposal” and the transition to the next stage is empty then the total weight for this example is 2.2. This means that stage “Union3 proposal” stage will be selected with probability 1.2/2.2. Stage “management 3 response” will be selected with probability 1/2.2.

3. If all expressions for stage transitions are not integer, real or empty expressions, then the program calculates the expressions in the order. It calculates ‘Boolean” (or it transforms the expression into Boolean if it is at all possible).
   a. For example text “true” is a text and it can be transformed into Boolean and therefore the expression receives the value “true”.
   b. Also number can be transformed into Boolean. Convention comes from the computer programming and says that the value 0 is “false”, all the other numbers are “true”.
   c. The next “exception” is an empty space which is interpreted as “true”.

In order to determine the way to move from the current stage to any of the other stages, the conditions for the transition have to be stated.

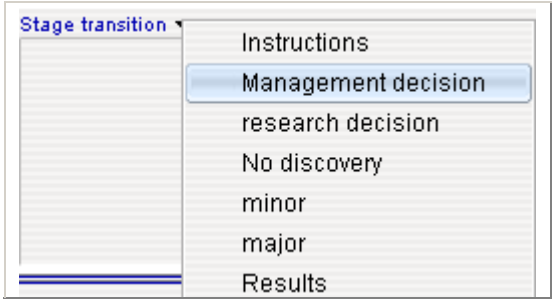
4. Selecting a stage transition

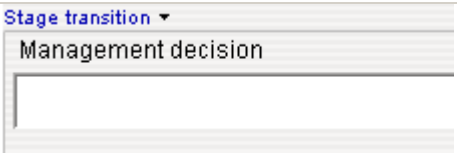
   a. By clicking on the arrow next to the Stage transition, all the states that are defined are listed. 

.

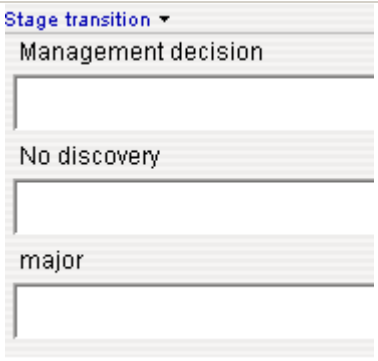
   b. Select a stage in order to define the condition for the transition.

Click on that stage and it will appear under stage transition:

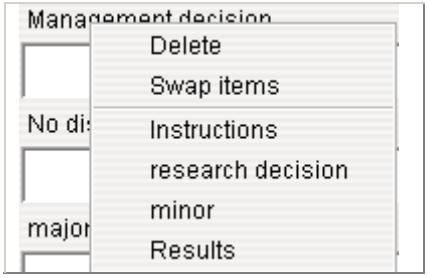




c. To add another stage, select the appropriate stage from the list of stages (i.e. if the stage is currently not create it, you should create it and it will be automatically append to the list of existing stages) and it will be appear below the existing selected stage.



5. Delete a stage from the stage transition



Select the stage to delete by right clicking on the name of the stage that should be deleted and click on "Delete".

In the example 'Management decision' was selected to be deleted.

3. Determining conditions to move to the next stage

To determine the condition to move to the next stage leave the expression empty or write a Boolean expression.

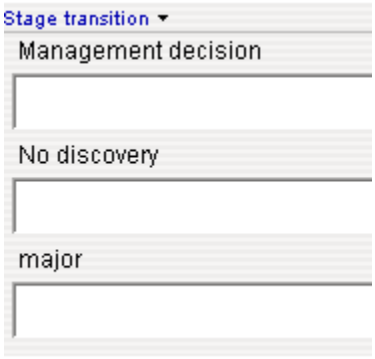
a. One stage transition

It means that transition from the current stage to the next stage occurs with probability 1. By default we can leave it blank or write 1.

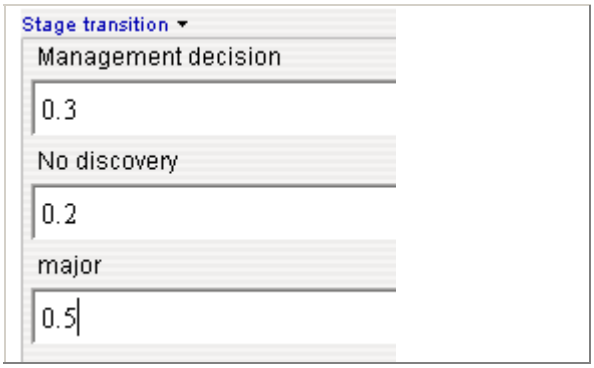
In this example the next stage that a player will see is a "shareholder offer" stage.

b. Equal probability/weight for stage transitions.

Empty expressions in the stage transitions that were selected are equivalent to writing the weight of 1. It means that transition from the current stage to any of the selected stages occurs with equal probability/weight. In the example below with probability 1/3 “Management decision” stage will be reached, with probability 1/3 “no discovery” stage will be reached and with probability 1/3 “major” stage will be reached.



c. Different probability/weight for stage transitions



Creating markets

1. Creating an asset

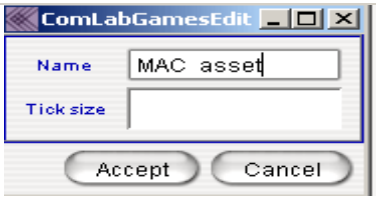
a. To create a market drag and drop an 

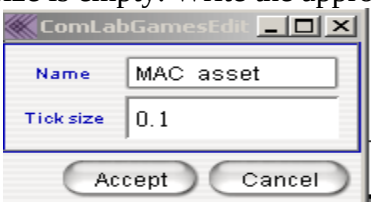
into the design window. One 

icon represents one market. To change the default name 

write the new name on the icon 

b. Double click on the created icon to change the tick size. When the asset window

opens 

the Tick size is empty. Write the appropriate minimum unit bids or asks can be submitted. 

In this example the minimum tick size was set to 0.1.

2. Determining characteristics of a created asset

a. In order to determine the endowment of each player type, information observed during the trading period, which information will be plotted or extracted from trading you have to create a 

and 

.

b. If only 

is created then player type editor does not have any information about the assets and it has the following format:

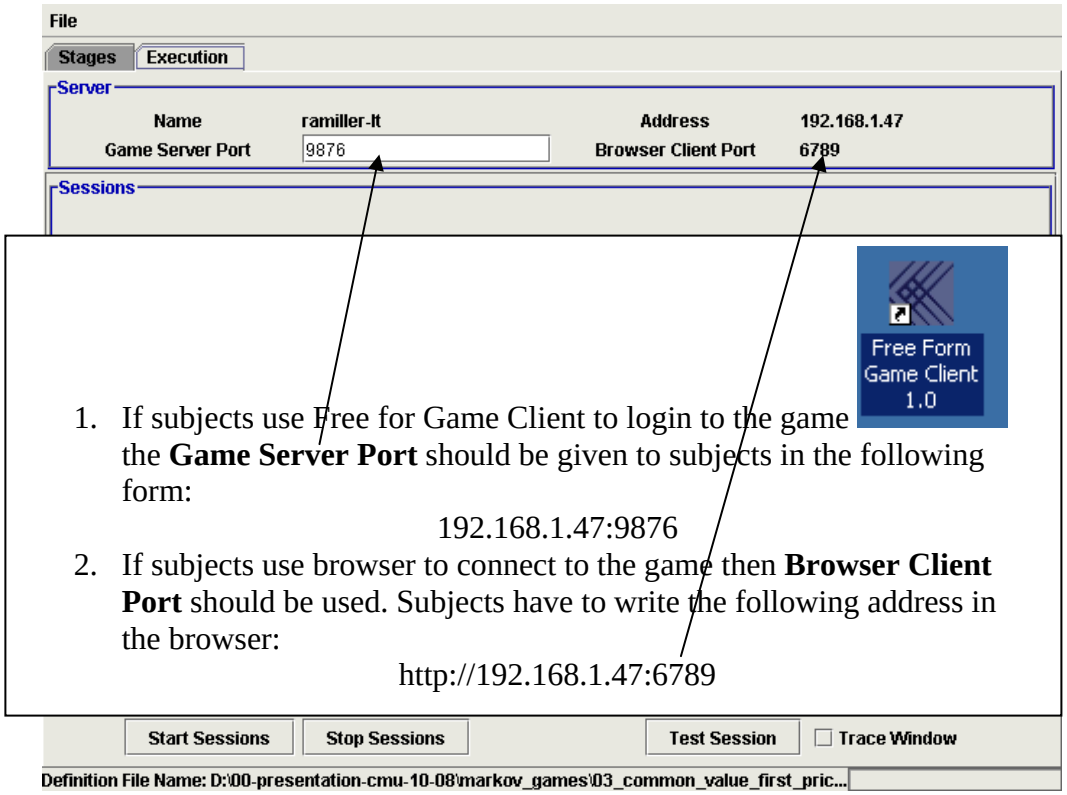
c. If only 

is created then you cannot play the game

Running an experiment

a. Click on the execution page and the window with the information about the server address appears as well as the option to start the game (i.e. Start Sessions)or to Test

sessions:



1. If subjects use Free for Game Client to login to the game the **Game Server Port** should be given to subjects in the following form:

   192.168.1.47:9876

2. If subjects use browser to connect to the game then **Browser Client Port** should be used. Subjects have to write the following address in the browser:

   http://192.168.1.47:6789

b. Before clicking Start Sessions it is useful to Test the game by clicking on Test Session and using Trace Window option. Although Trace window option slows the program it shows all the steps in calculating the state variables and how the program moves from one stage to the next.

- Check Trace Window and then click on Test session:

- Below is an example of the trace window with all appropriate information about the variables, choices, etc.:

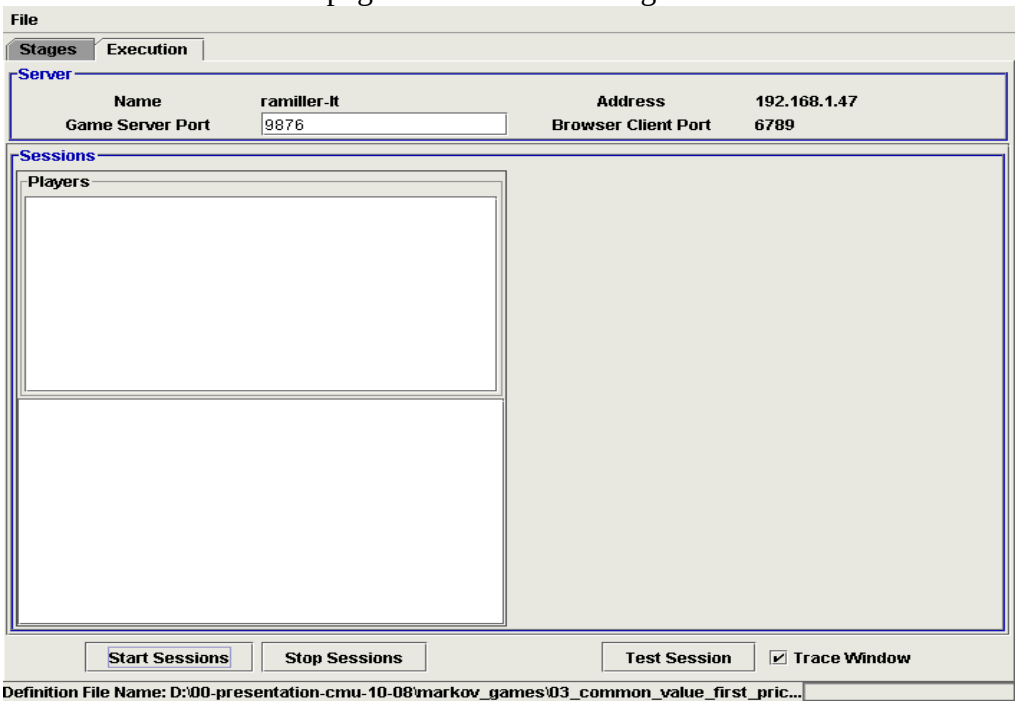
If there is mistake in the program it will show where it occurs.

- In addition to the trace window the client windows appear for a moderator to go through the whole experiment and observe how the subject will see it.

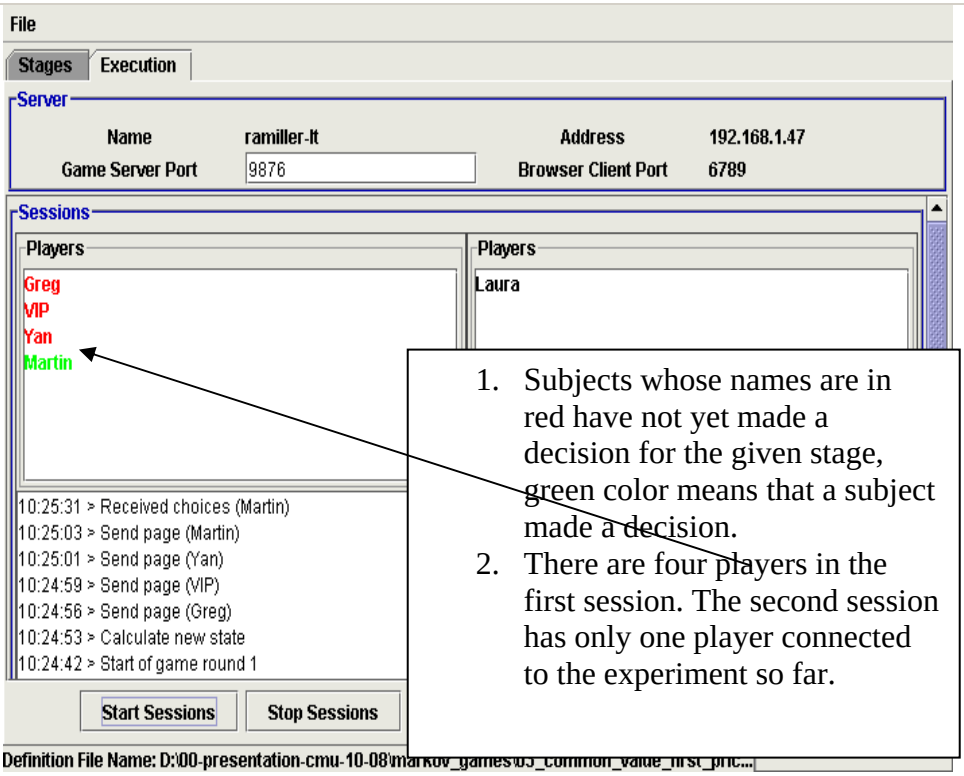
c. Once the test is done proceed to running the experiment by clicking on 

.

d. The moderator Execution page will look like the figure below:



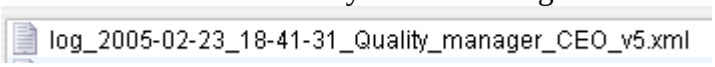
e. Once subjects connect to the game, the moderator will be able to see who is connect to a given session and if they made a decision in a given stage:



Viewing experimental results

a. The output file is automatically created when a moderator clicks on 

. The file name has always the following form:



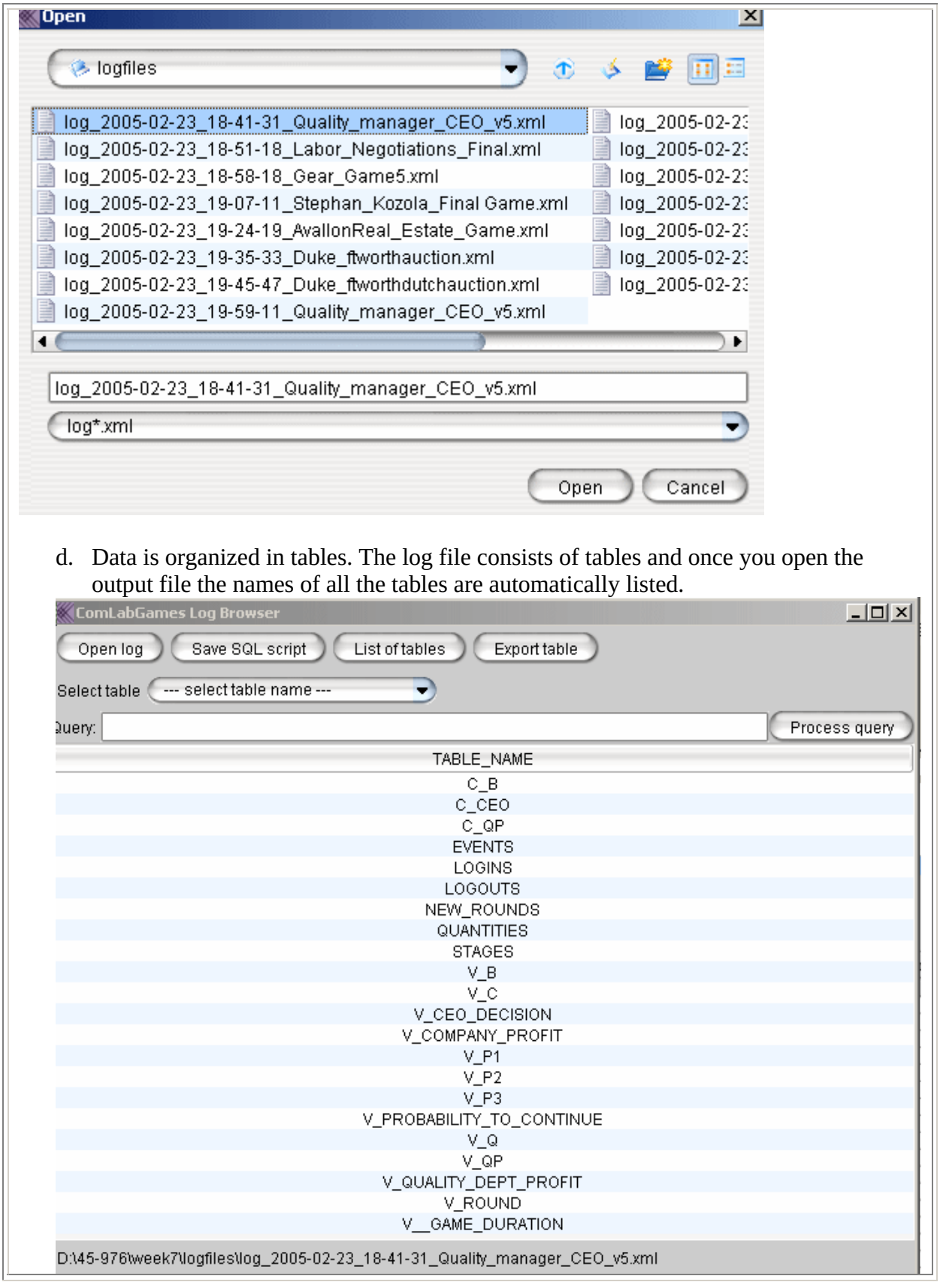
- start with the log, date, time when the session started and the name of the file. This enables moderator to know exactly when the same game was played on different occasions.

b. To open the file and view the results click on the **Comlabgames Log Browser-Free icon** 

that is on your desktop and it was downloaded from the Comlabgames web page.

c. Open the file that you want to look by clicking on 

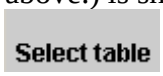
and selecting the appropriate file:



d. Data is organized in tables. The log file consists of tables and once you open the output file the names of all the tables are automatically listed.

e. By clicking on 

the list of tables that are created (see the in the figure above.) is shown.

f. 

allows to select a table and list the data related to this specific table.

g. 

can be used to organize and display the data. SQL commands can be used to filter the data in certain way.

Each 

created in the design window represents a separate table. All choices start with capital C_ and continue with the name of the choice created in the program. For example table

TABLE_NAME
C_B

represents a choice 

that you see on the design window when you open a game file.

h. Each variable created in the design window is also presented in a separate table. All variables start with capital V_ and continue with the name of the variable created in the program. For example table

TABLE_NAME
V_COMPANY_PROFIT

represents a variable 

that you see on the design window when you open a file with which you will run an experiment.

i. The following tables:

TABLE_NAME
EVENTS
LOGINS
LOGOUTS
NEW_ROUNDS

have designated name are always created as a part of output with LogBrowser.

- Table “Events” has all the data recorded but the data are appended by rows not merged by columns. Events table lists all events during the experiment.
- “Logins” capture all the successful logins to the program by user name. Similarly for logouts.
- “New_rounds” contains the data about the new round.

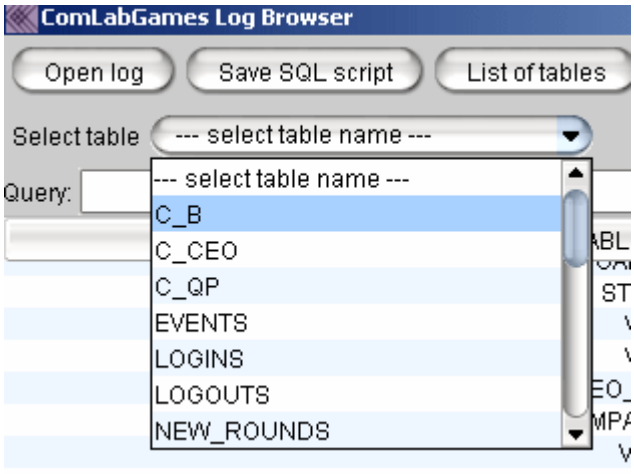
j. The amount of information from the experiment is determined by the moderator. It is up to the moderator to decide which kind of variables he/she wants to create (even if those variables are not displayed to subjects) in order to find them in the log file.

Two ways of selecting a table:

I. **Select table** enables presentation of the data for the selected table by clicking on the table name. Click on the drop down arrow next to **Select table** and all the names of the tables that are contained in the file are listed. Click on the name of the table to view it.

In our example the choice C_B was selected from

Select table --- select table name ---.



After selecting table C_B the following information about choice b appears automatically in the LogBrowser:

TIME	SESSION_ID	ROUND	PLAYER_NA...	SELECTED_...	SELECTED_...	ROW	COLUMN_1
18:45:35.311	2	1	Di Wei	-1	-1	1	5
18:46:20.135	1	1	Jason P.	-1	-1	1	5
18:47:04.549	2	1	Di Wei	-1	-1	1	5
18:47:07.383	3	1	hucks	-1	-1	1	3
18:47:36.315	1	1	Jason P.	-1	-1	1	3.5
18:47:44.546	2	1	Di Wei	-1	-1	1	4
18:48:24.023	6	1	hucks	-1	-1	1	3.5
18:50:30.205	4	1	imm	-1	-1	1	4
18:51:10.933	5	1	Karl Hafner	-1	-1	1	3

b. Saving the selected table in ASCII format

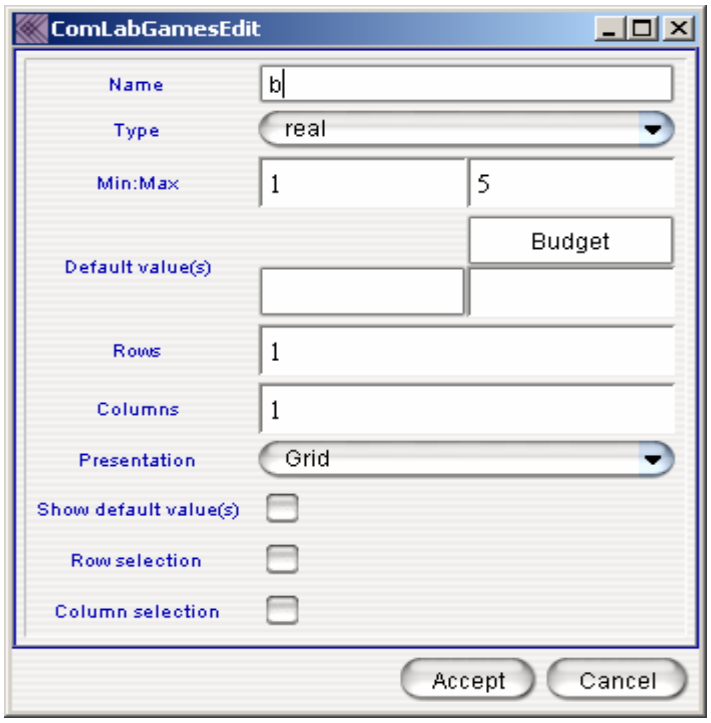
Click on 

to export the information about the table

b. Presentation of the data in the table

Once the table is selected all the data for this choice that are captured by the program are listed in the columns. It always starts with the time a choice was made, the session_id that represents a number player is assigned to, round number, player name which represents the login name, selected_row if you specified row selection in the choice editor (see the choice editor below and the data presentation;the value will be -1 if you did not use this option), selected_column if you specified Column selection in the choice editor the row number (the value will be -1 if you did not use this option) and under Column_1 the amount that they wrote for b (i.e. budget) is shown. 

was defined as



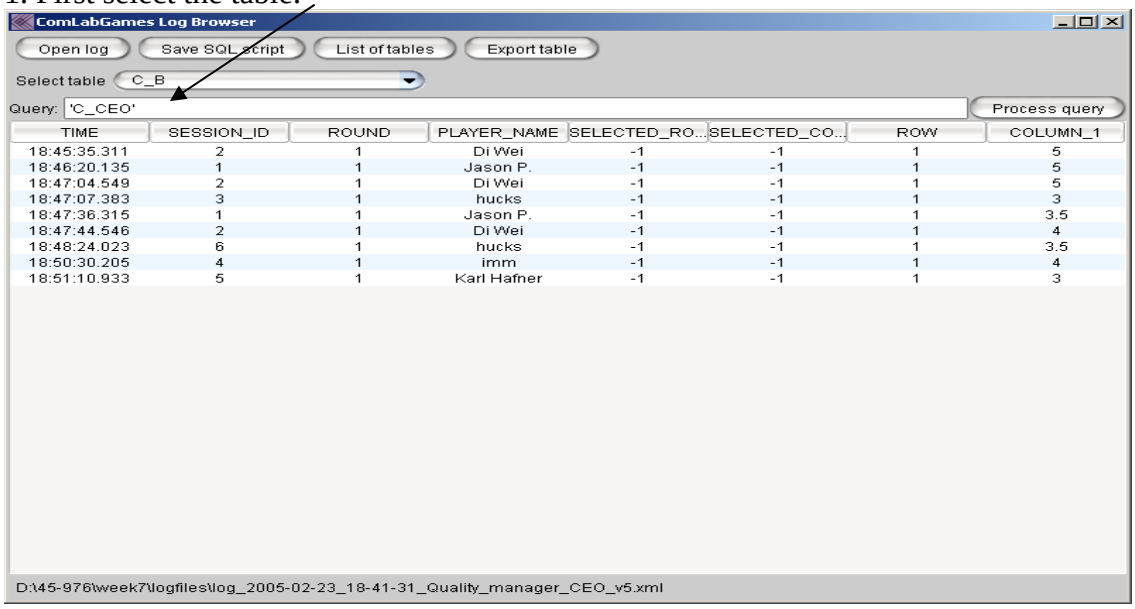
numbers in Column_1 represent subject's budget choices.

Once you selected the table just click on 

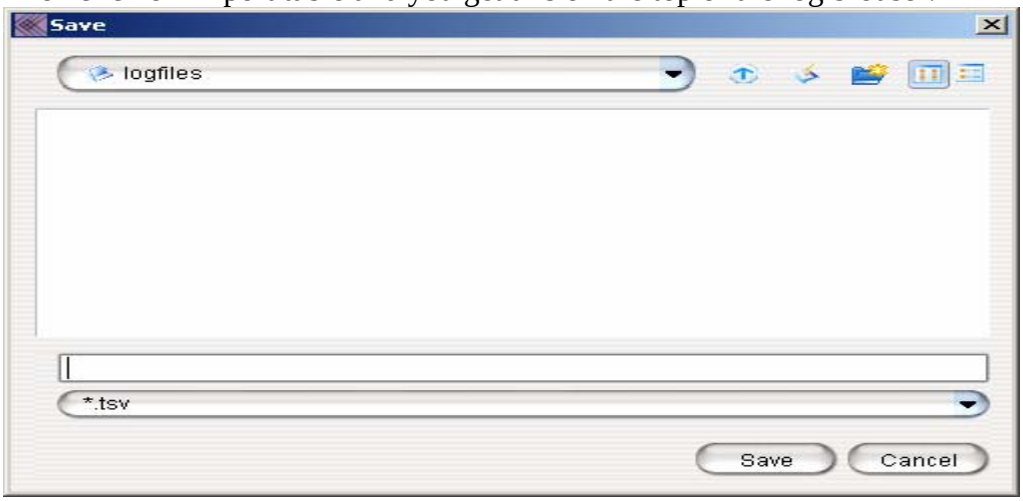
and you will get the following

message:

1. First select the table.



Then click on Export table and you get this on the top of the log brouser:

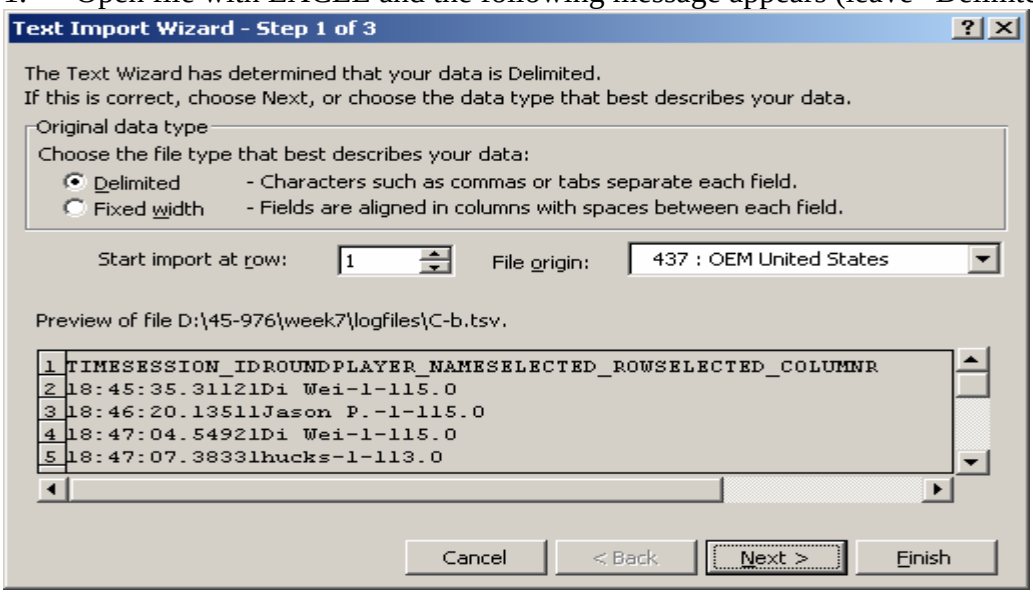


Then write (the same name as the table):

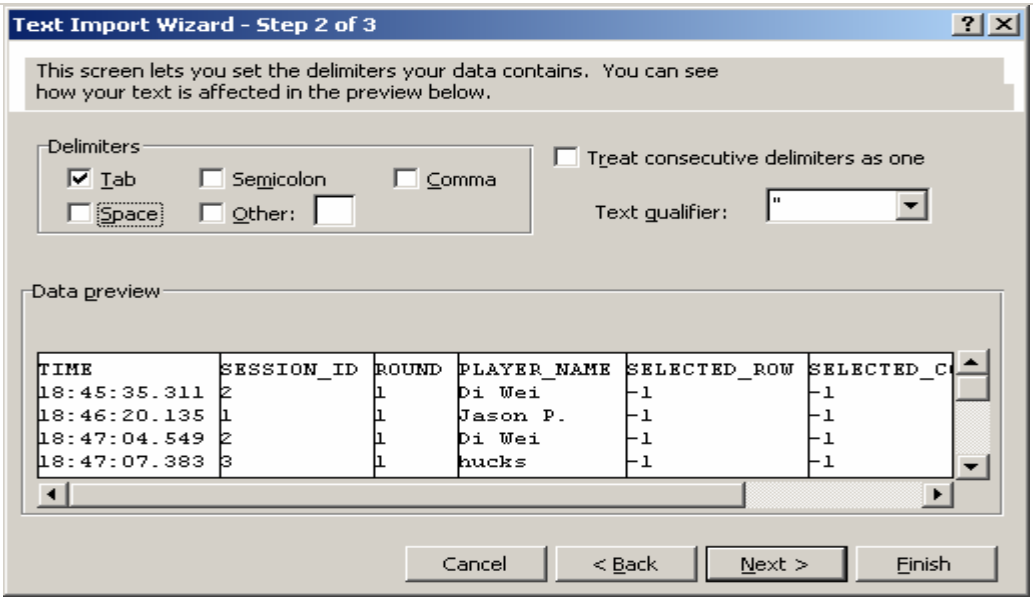
When you go to Excel file, open all files and this particular file will be called: **C_b.tsv.** This files are “ASCII” files.

Opening a created ASCII log file with Excel

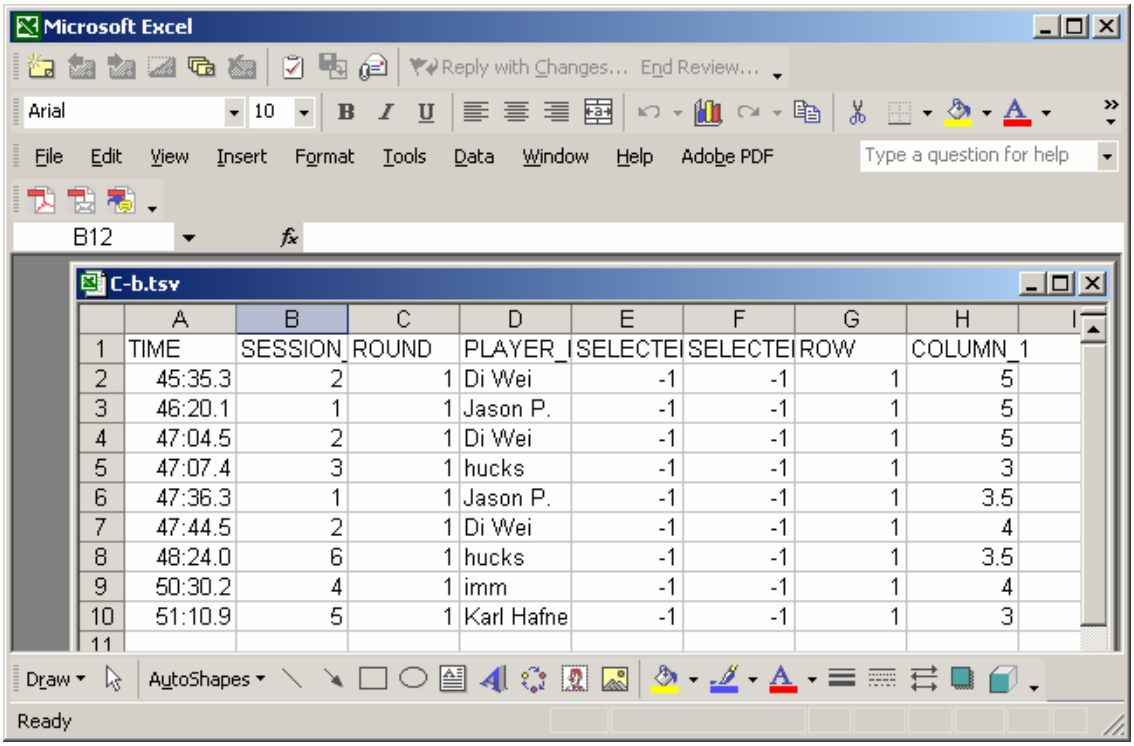
1. Open file with EXCEL and the following message appears (leave “Delimted”):



2. Click Next and the following windows appear:



3. Leave "Tab" delimiters and click Next and then Finish and the table is imported into Excel:

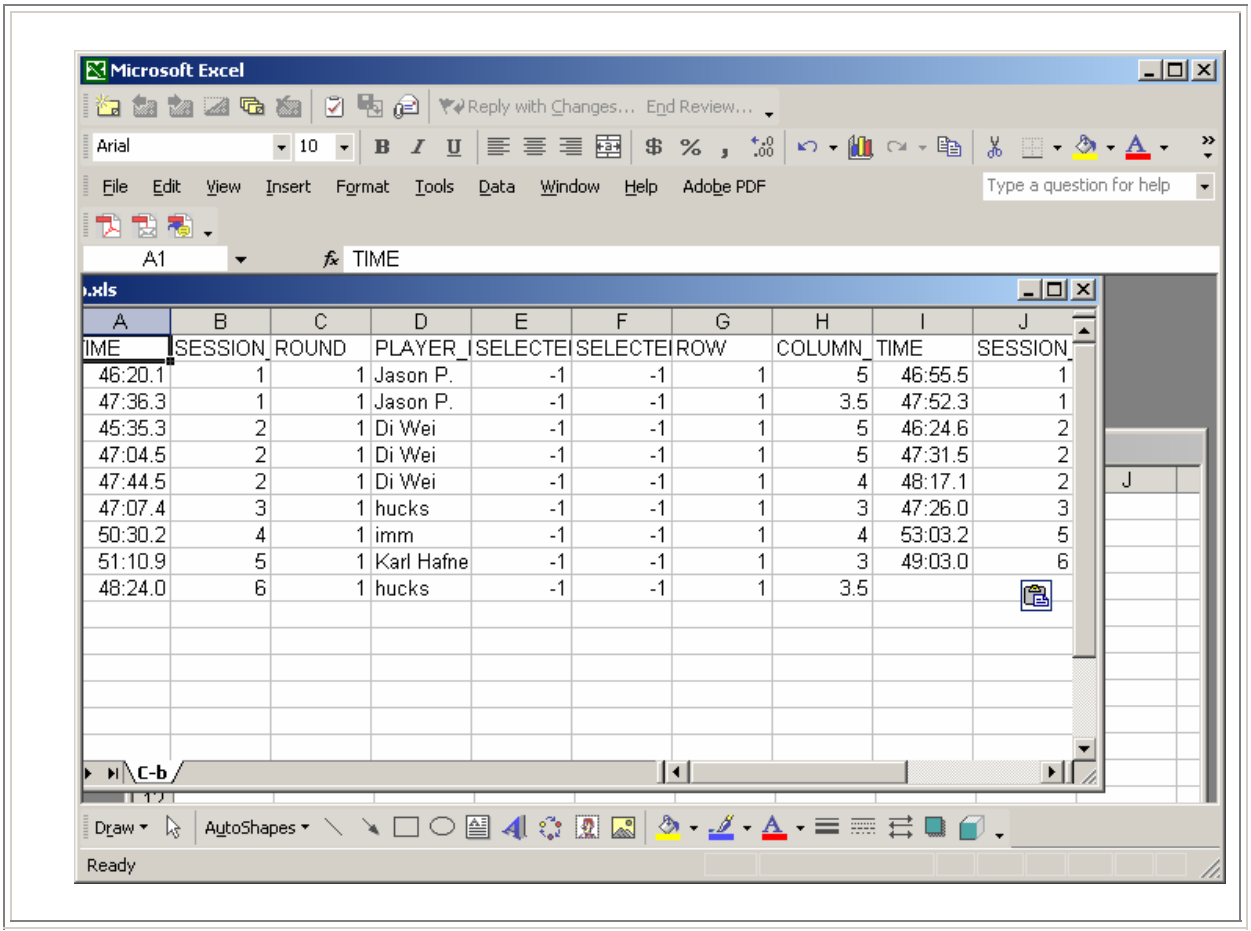


	A	B	C	D	E	F	G	H
1	TIME	SESSION_	ROUND	PLAYER_I	SELECTEI	SELECTEI	ROW	COLUMN_1
2	45:35.3	2	1	Di Wei	-1	-1	1	5
3	46:20.1	1	1	Jason P.	-1	-1	1	5
4	47:04.5	2	1	Di Wei	-1	-1	1	5
5	47:07.4	3	1	hucks	-1	-1	1	3
6	47:36.3	1	1	Jason P.	-1	-1	1	3.5
7	47:44.5	2	1	Di Wei	-1	-1	1	4
8	48:24.0	6	1	hucks	-1	-1	1	3.5
9	50:30.2	4	1	imm	-1	-1	1	4
10	51:10.9	5	1	Karl Hafne	-1	-1	1	3
11								

Joining tables together in Excel:

1. Export the other tables (i.e. choices, variables) one by one the way we did it in the example above.

2. Paste this table in the column next to the original table:

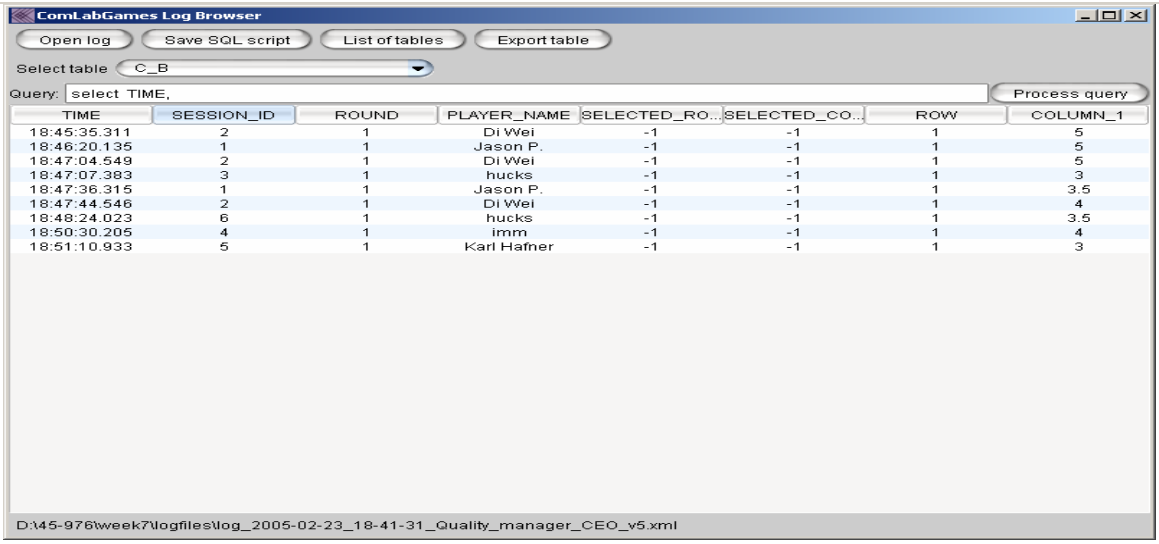


A	B	C	D	E	F	G	H	I	J
IME	SESSION_	ROUND	PLAYER_I	SELECTEI	SELECTEI	ROW	COLUMN_	TIME	SESSION_
46:20.1	1	1	Jason P.	-1	-1	1	5	46:55.5	1
47:36.3	1	1	Jason P.	-1	-1	1	3.5	47:52.3	1
45:35.3	2	1	Di Wei	-1	-1	1	5	46:24.6	2
47:04.5	2	1	Di Wei	-1	-1	1	5	47:31.5	2
47:44.5	2	1	Di Wei	-1	-1	1	4	48:17.1	2
47:07.4	3	1	hucks	-1	-1	1	3	47:26.0	3
50:30.2	4	1	imm	-1	-1	1	4	53:03.2	5
51:10.9	5	1	Karl Hafne	-1	-1	1	3	49:03.0	6
48:24.0	6	1	hucks	-1	-1	1	3.5		

More advanced presentation of tables in the log file:

If you are more sophisticated:

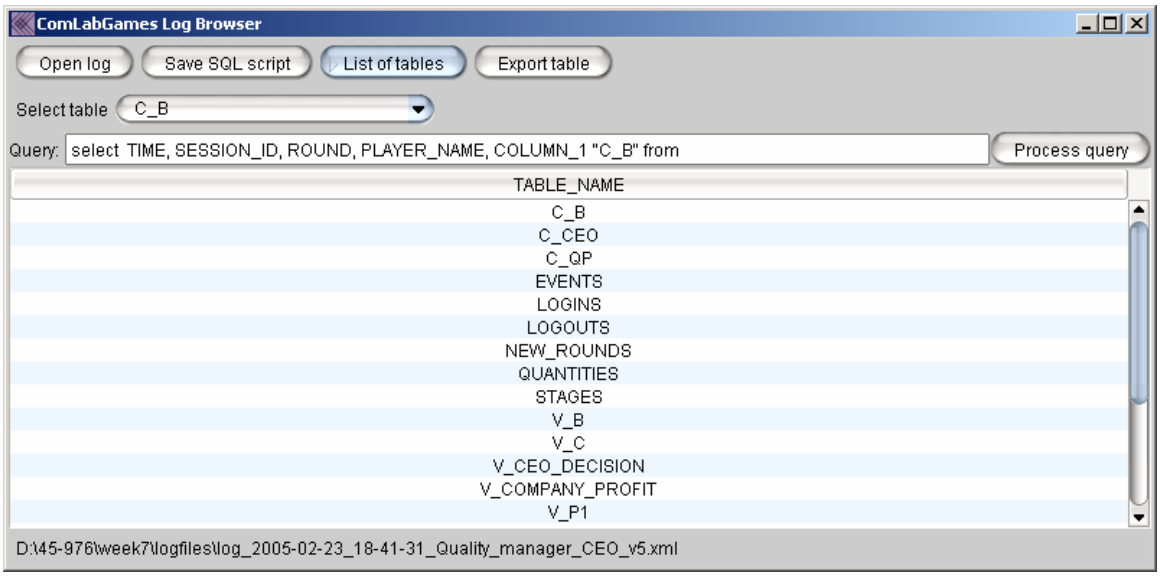
Type select in query, then double click on any thing you want to select. Double click on time, then continue double clicking on any heading in the table that you want to select (put comma in between)



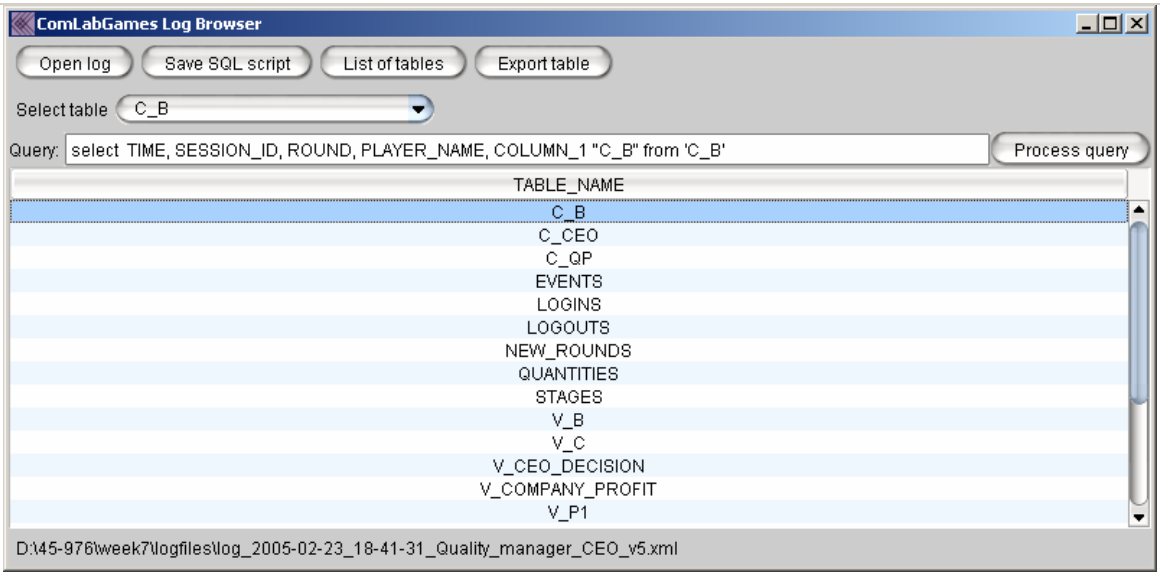
At the end the following items appeared in the query:
select TIME, SESSION_ID, ROUND, PLAYER_NAME, COLUMN_1 “C_B” from

Note after double clicking on COLUMN_1 “C_B” was written to rename COLUMN_1 to the choice called C_B. Also from was written.

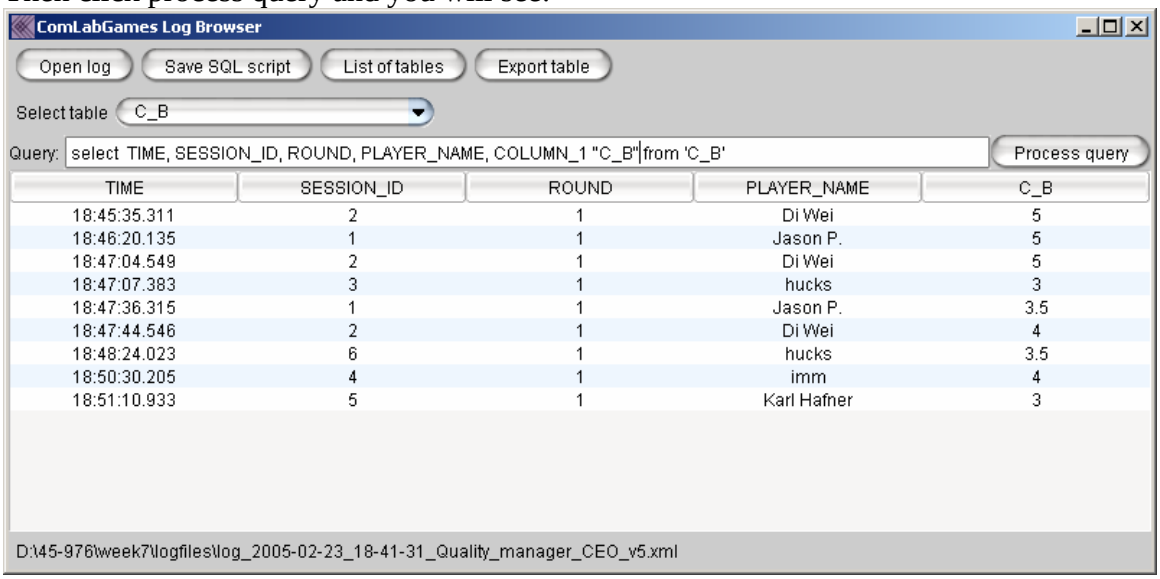
then click on list tables and you will see the following. Double click on C_B because you want the data from that table.



You will see the following:



Then click process query and you will see:

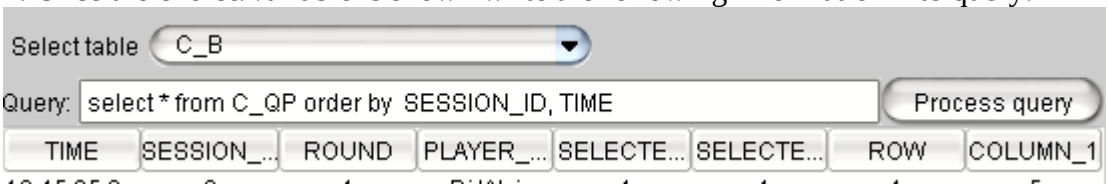


Then do the usual export table. As you see you were able to label the variables/choices. Instead of column_1 you have C_B by doing column_1 “C_B”

Ordering data in log browser before exporting

1.Open the file and select the appropriate choice/variable:

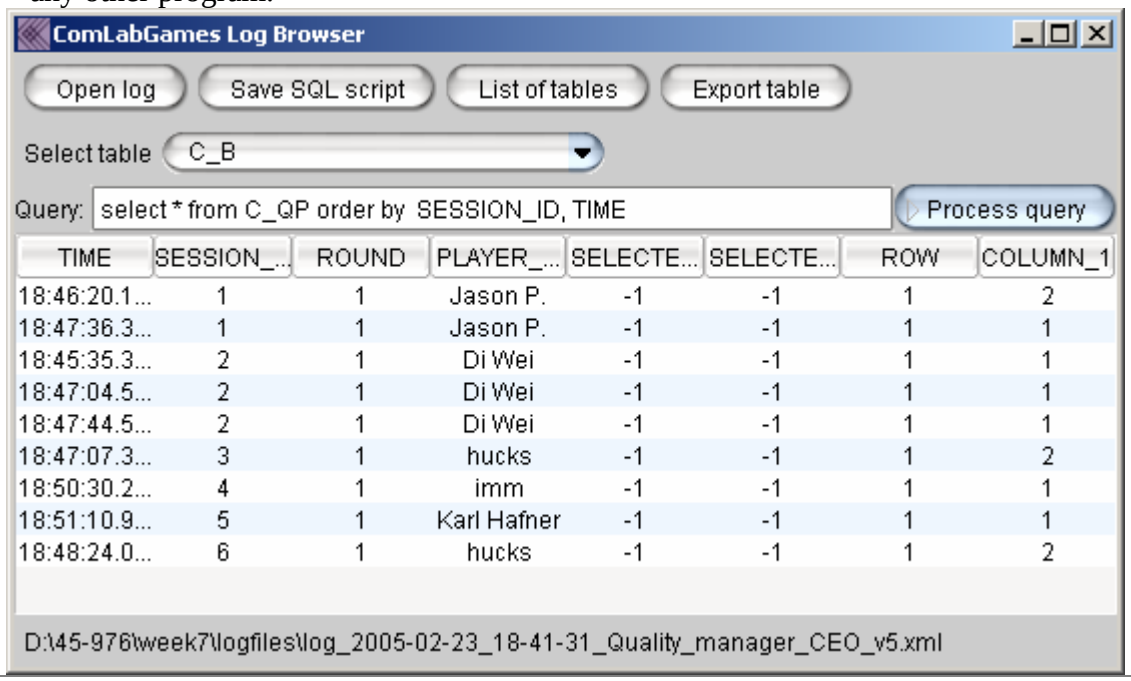
2. Once the choice/variable is shown write the following information into query:



If you want to order it by different criteria, separate it by comma.

Note you do not need to write SESSION_ID, TIME. You can just double click on the heading of the columns and it brings it up into Query.

3. The data get sorted and if you export this table it will appear the same way in Excel or any other program.



Joining data inside LogBrowser before exporting

1. There exists several ways to join the data:

a. SELECT * FROM table1, table2

Example: If we in one table data:

John 170cm
Mary 165cm

And in the other table data:

John book $30
John calculator $50
Mary book $20
Mary CD $25

Then SELECT * FROM table1, table2 does the following:

John 170cm John book $30
John 170cm John calculator $50
John 170cm Mary book $20
John 170cm Mary CD $25
Mary 165cm John book $30
Mary 165cm John calculator $50
Mary 165cm Mary book $20
Mary 165cm Mary CD $25

b. SELECT * FROM table1, table2 WHERE table1.name=table2.name gives the following result:

John 170cm John book $30
John 170cm John calculator $50
Mary 165cm Mary book $20
Mary 165cm Mary CD $25

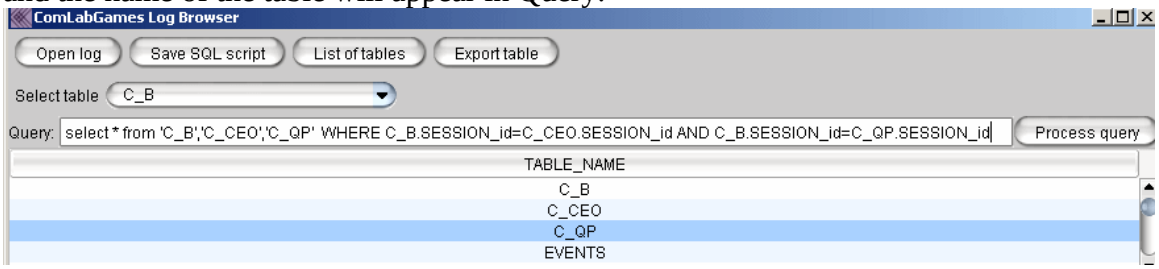
An example of joining the tables using the log data:

1. First click on “List of tables” and the list of tables appear
2. Write the following statement into Query:

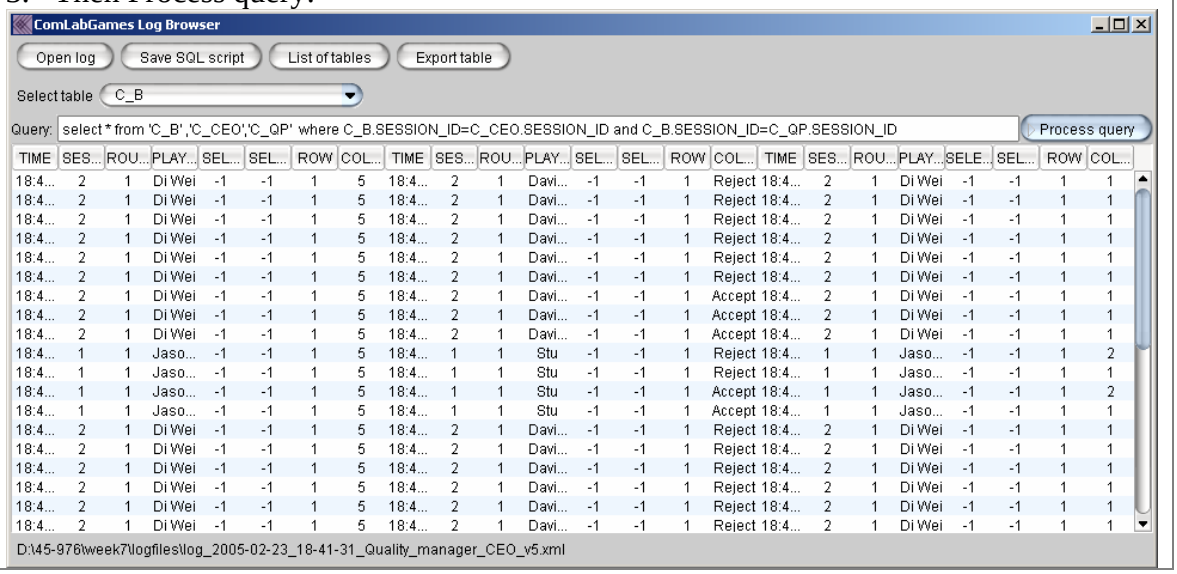
select * from 'C_B','C_CEO','C_QP' WHERE C_B.SESSION_id=C_CEO.SESSION_id AND C_B.SESSION_id=C_QP.SESSION_id

Note that you do not need to write ‘C_B’ just double click on the appropriate table below

and the name of the table will appear in Query.



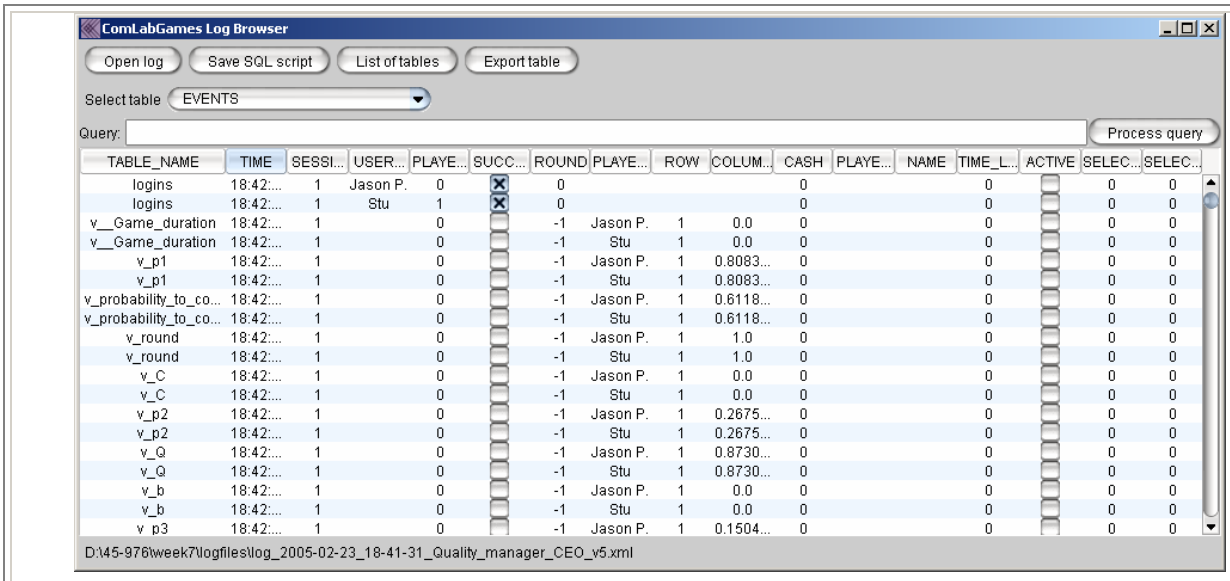
3. Then Process query:



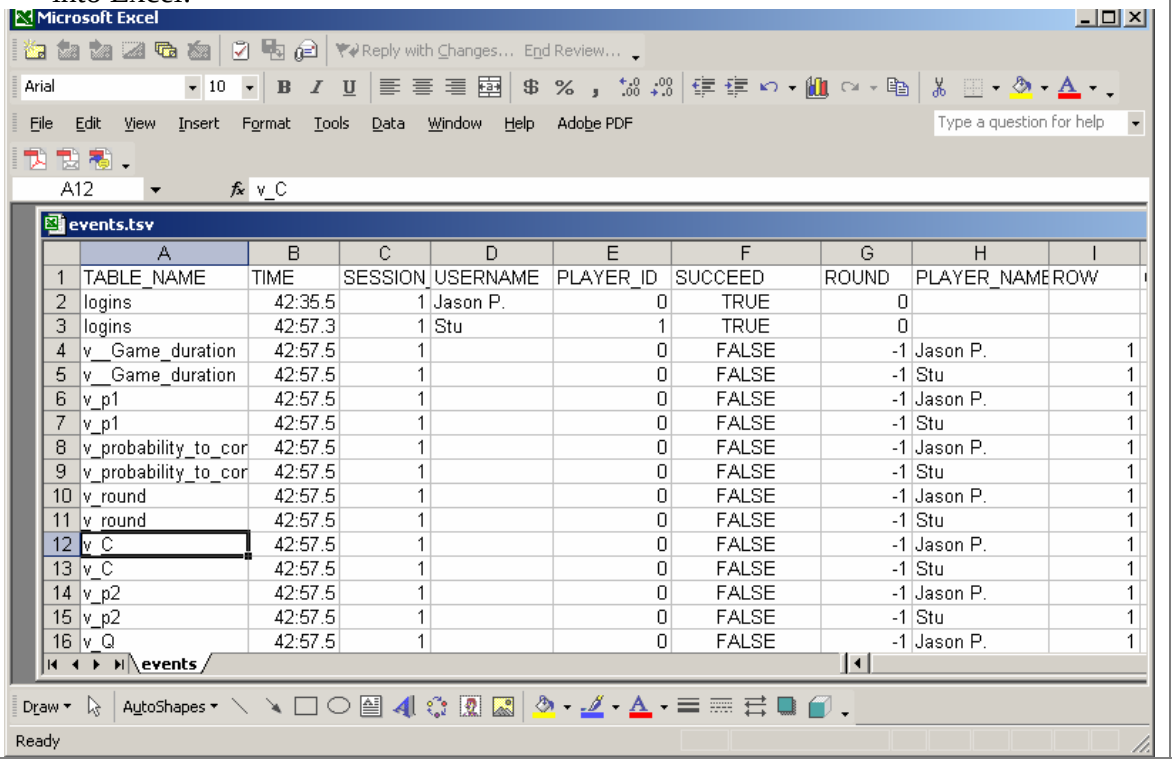
Exporting all the data that are recorded in the Events table:

1. When you open log browser, select Events table.

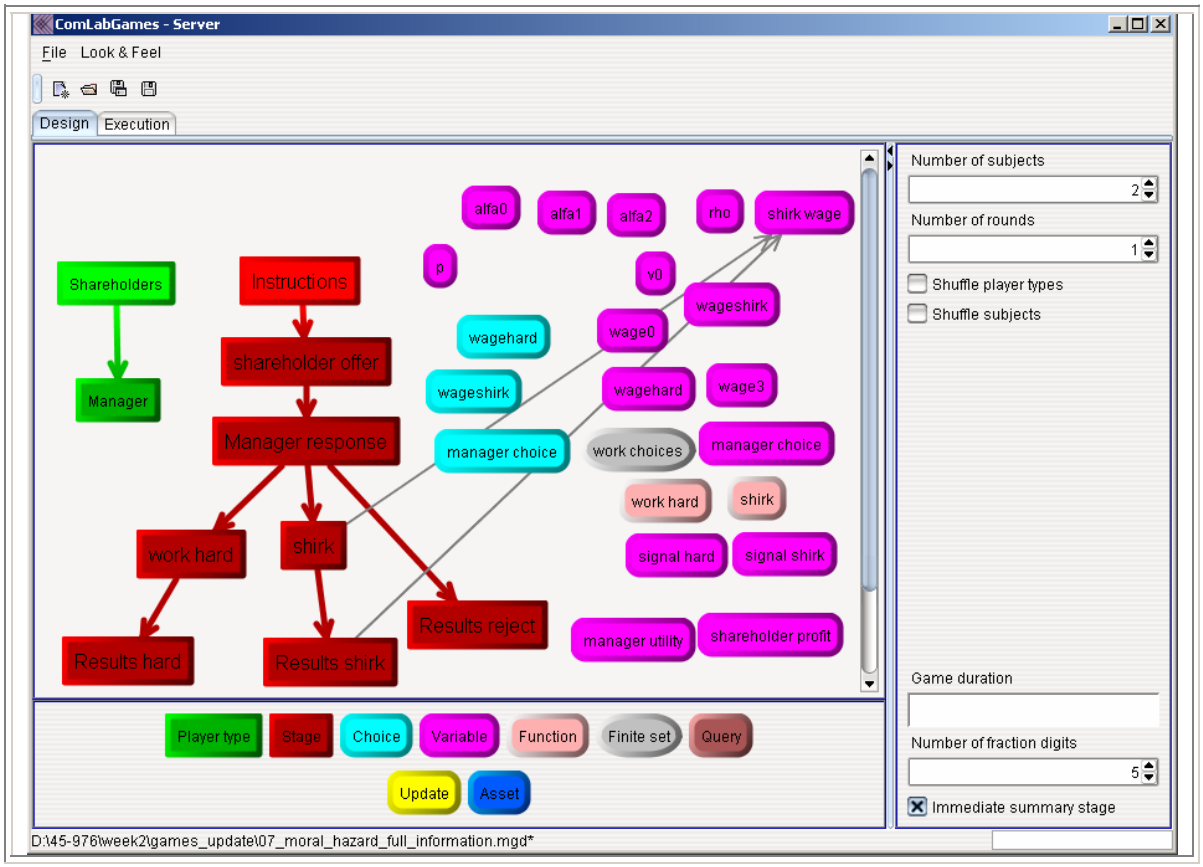
The data are here recording in the orders that happened during the experiment. All the choices, variables are listed under table_name. See below how it appears in the log browser.



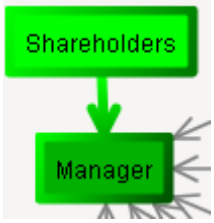
2. Expert this table and then import it into Excel in the same way we opened other tables into Excel:



I. An example of creating an experiment in Free form.



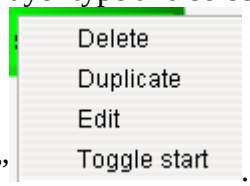
1. Creating player types:



Important:

a. One of the player types has to be defined as the starting point from which the conditions are checked. By default the first player that is created is where the program starts checking. The starting player type has a light green color.

To change the starting player from, right click on the player type and select



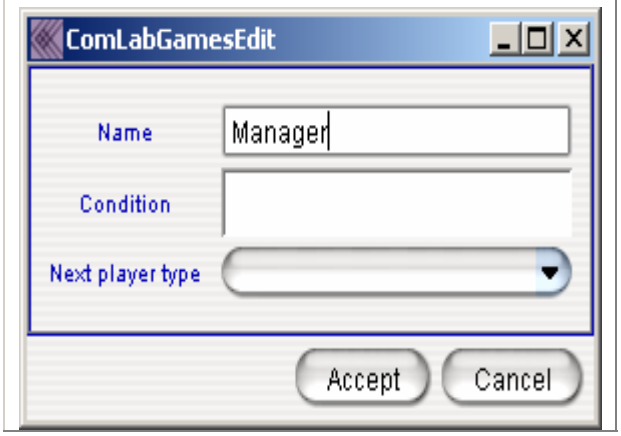
"Toggle start".

b. All conditions are Boolean.

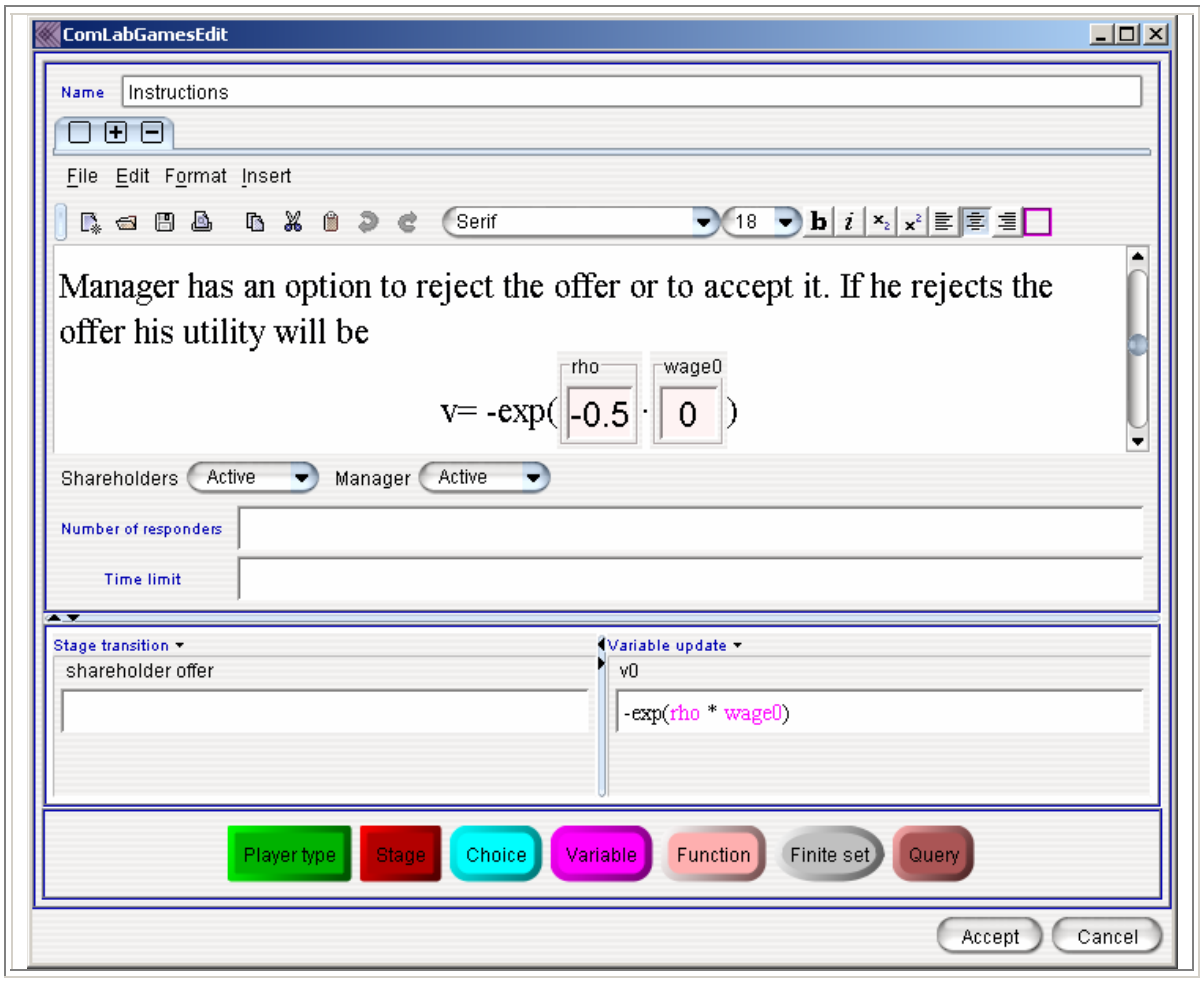
Shareholders is the starting player and we defined that the first subject who logon to the program is assigned to be a shareholder. Subject_number()=1.

If the condition is not satisfied then the next player type to check the condition is "Manager"

Manager does not have any condition specified and neither the "Next player type". There are only two player types and if the condition for Shareholders is not satisfied then the Manager role will be given to the next player who connects to the game



2. Instructions page

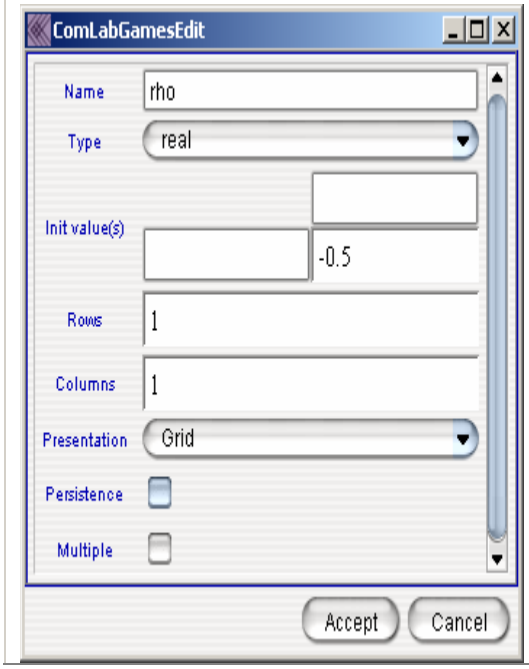


a. In the instructions several variables are inserted like 

. The list of all the variables that are inserted in the text is:

On the design window 

icon is created. By double clicking we see how it is defined: As a real number with an initial value of -0.5.



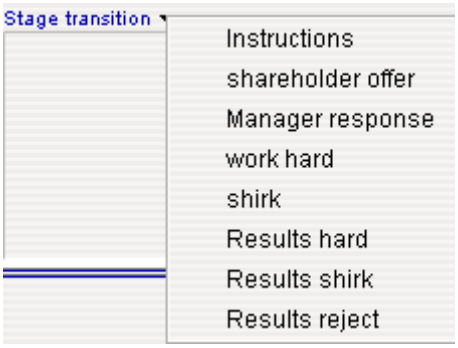
Similarly the following “constants” are defined:

3. Player’s task on instructions page

Both player types are reading the instructions and have to press the continue button in order to proceed to the next stage. Therefore shareholders and manager are defined as “Active”

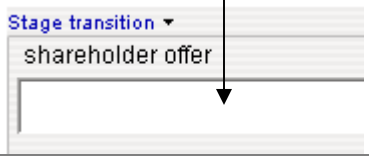
4. Transition on the instruction's stage

By clicking on the arrow next to the Stage transition , all the states that are defined are listed.



shareholder offer was selected to be the stage that follows the Instructions stage with probability 1.

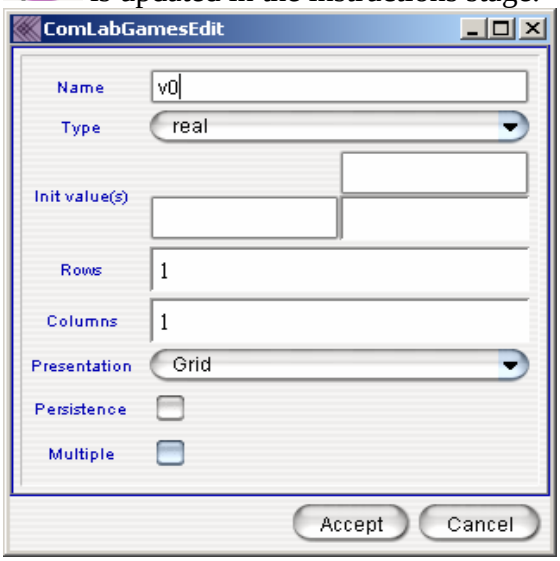
If the space is empty for the selected stage transition it means that with probability 1 subject will go to the next stage. In this case shareholder offer



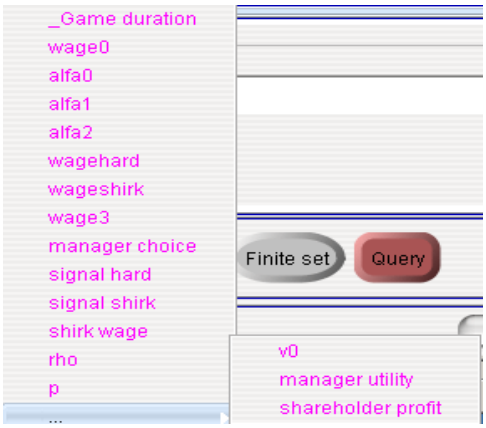
5. Update of the variables on the instruction stage

Type of v0 variable is real. No initial value is defined.

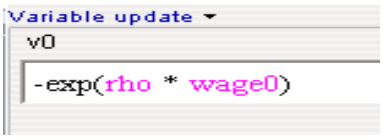
v0 is updated in the instructions stage.



To update the value for a variable click on the arrow next to Variable update and select the variable you want to update.



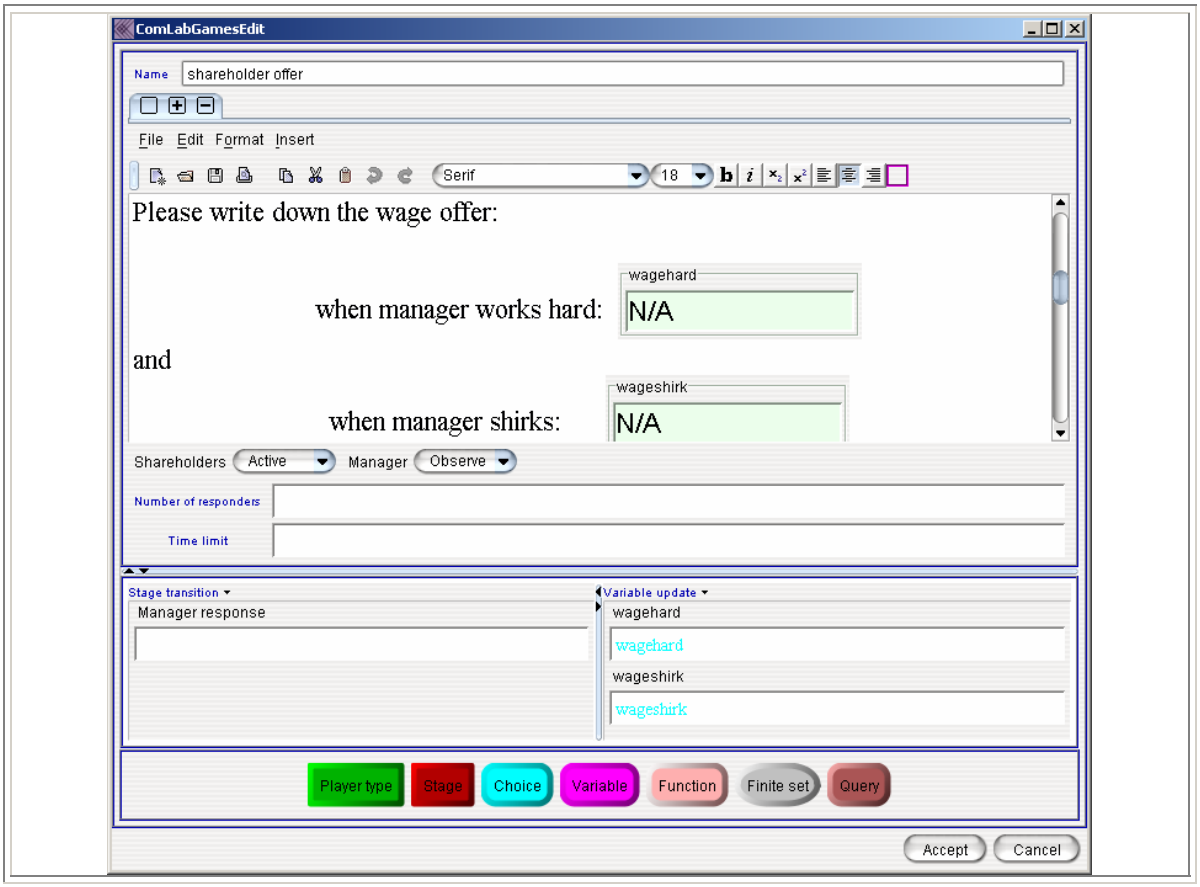
v0 was updated with the expression below.



Note that you can recall the assistant editor with "Alt"+"double click". All variables have to be written in

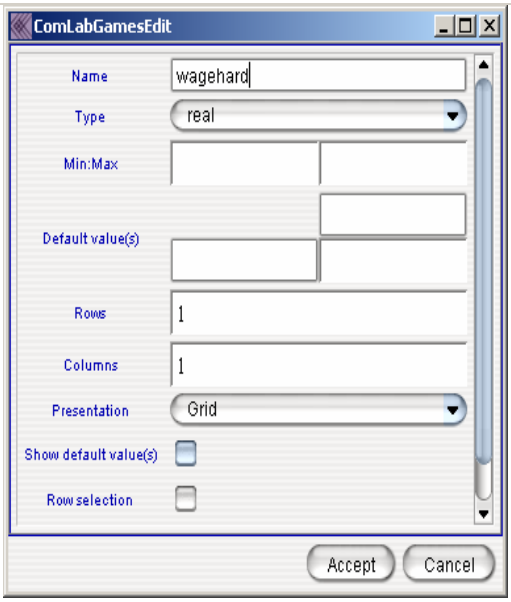
	purple (use “Crtl”+””Shift”+”V” to change the color of letter to purple. “V” stands for variable. “Crtl”+””Shift”+”N” changes the color of the letter to black. “N” stands for normal. “Crtl”+””Shift”+”C” changes the color of letter to turquoise. “C” stands for choice.

6. shareholder offer **stage**



7. Defining and inserting choices into shareholder offer **stage**

wagehard choice is defined as real with no minimum or maximum limit and neither in the default value defined.	Insert the choice into the text in the stage by clicking on Insert and selecting wagehard.

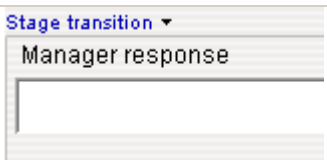


wageshirk is defined exactly the same way.

8. Player's task on shareholder offer **stage**

Shareholders are active. They have to make a decision about the wage they want to offer for working hard and the case of manager shirking. "Active" means that they have to press the continue button in order to proceed to the next stage.
The manager observes everything that shareholders sees except the "Continue" button is not activated and the manager cannot make decisions.

9. Stage transition on shareholder offer **stage**



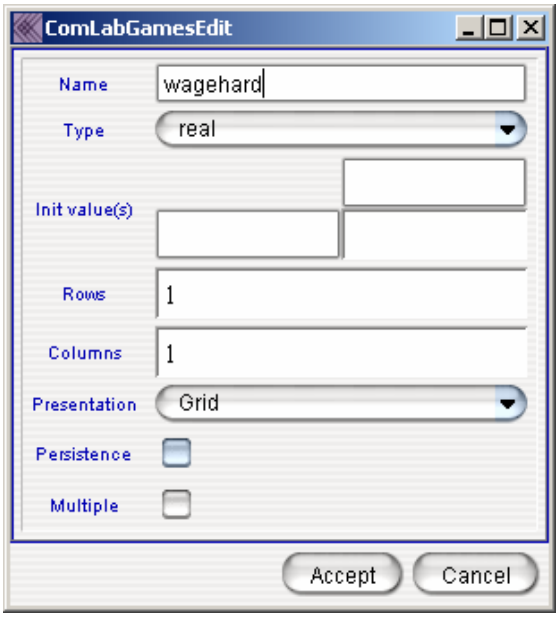
Manager response was selected to be the stage that follows the shareholder offer stage with probability 1.

10. Update of the variables on the shareholder offer **stage**

In order to display choices in later stages we have to define it as a variable.

Type of wagehard variable is real. No initial value is defined.

wagehard is updated in the shareholder offer stage because shareholder have to choose the wage when manager works hard. wagehard .



wagehard and wageshirk were updated with the wagehard choice and wageshirk choice.

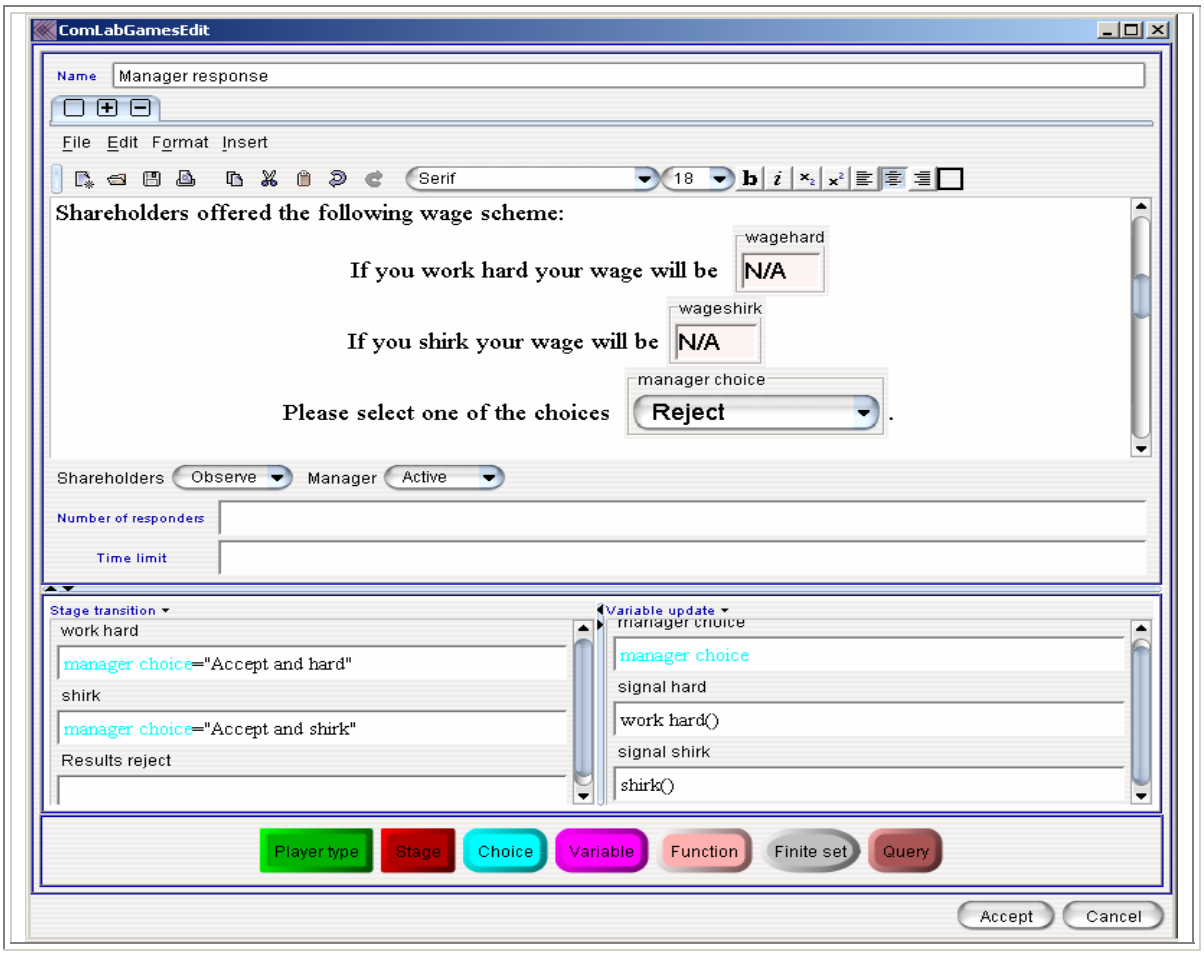
Note that you can recall the assistant editor with "Alt"+"double click" and insert the choice into the variable.

"Crtl"+""Shift"+"C" changes the color of letter to turquoise if you just type the choice. "C" stands for choice.

All variables have to be written in purple (use "Crtl"+""Shift"+"V" to change the color of letter to purple. "V" stands for variable.

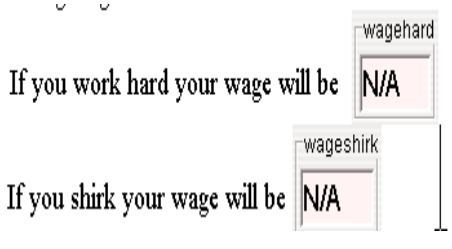
"Crtl"+""Shift"+"N" changes the color of the letter to black. "N" stands for normal.

11. Manager response **stage**



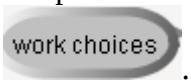
12. Inserting variables and choices into Manager response stage

The decisions that shareholders made in the shareholder offer stage are inserted as variables into the text so managers can observe shareholders' offers before making a decision.



Manager has a choice to reject the offer to accept and work hard or to accept and shirk. Manager's choice is defined as a finite set:

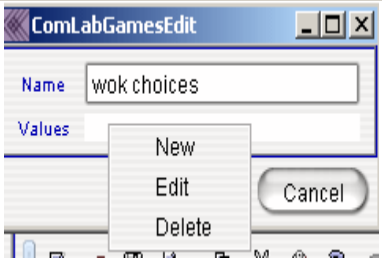
in the following way:

a. First define a finite set by dragging and dropping a finite set. In our example the finite choice was renamed to 

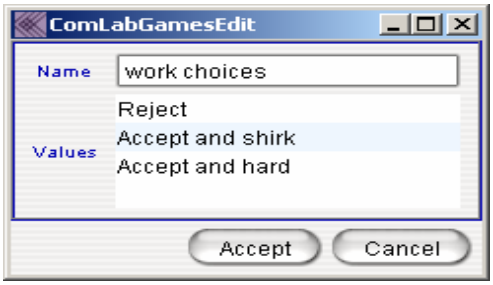
.

b. Double click on the 

and start defining the possible finite choices by right clicking on the space next to values. Select New for defining a new value.

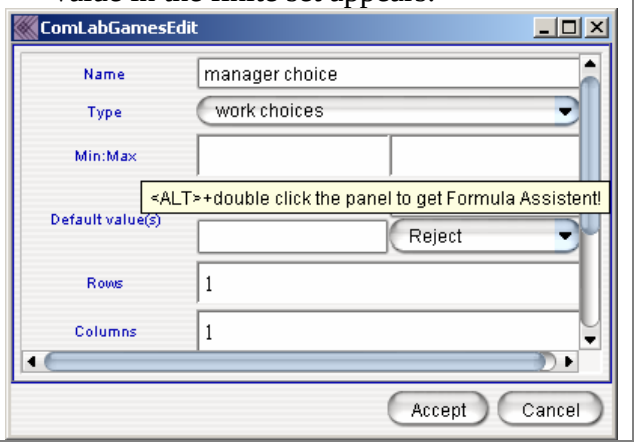


c. The following values were defined for work choices finite set:



d. Once the finite set is defined we can insert it into the 

by selecting work choice type. By default value the first value in the finite set appears.



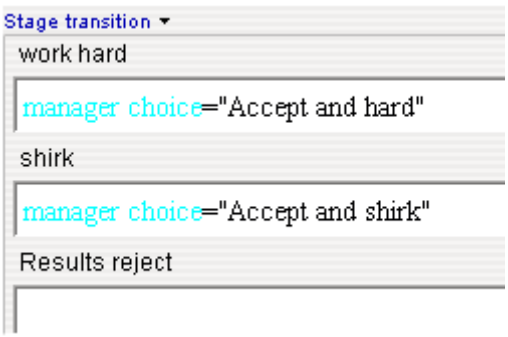
13. Player's task on 

stage

Now manager is active.
The shareholders observe everything that manager sees except the "Continue" button is not activated and the shareholders cannot make decisions.

14. Stage transition on 

stage



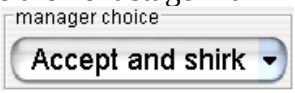
a. 

will be the next stage that follows the 

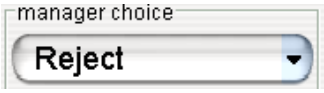
if a manager selects 

.

b. 

will be the next stage if a manager selects: 

b. 

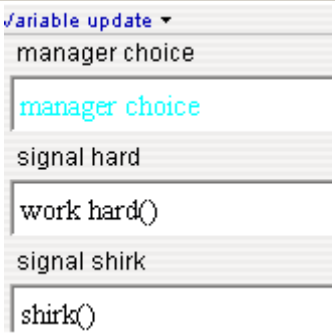
will be the next stage for any other choice that a manager makes. Note we can leave the stage transition empty in this case. Given that the only choice that was not selected is 

"Result reject" stage will be the next stage.

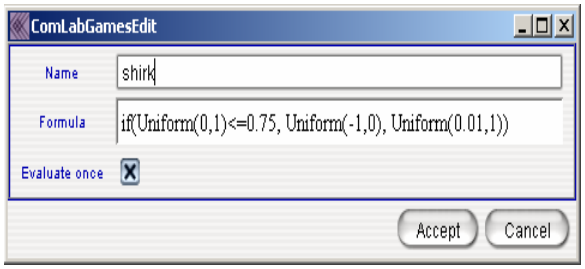
The choice condition has to be written in double quotation when a choice is a text or a finite set.

15. Update of the variables on the 

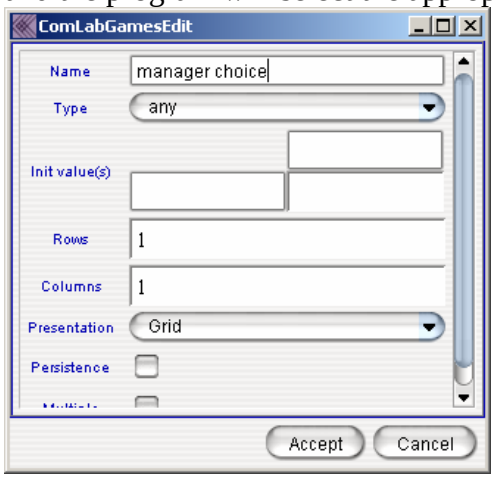
stage



Similarly we define signal shirk where the function shirk is defined as:

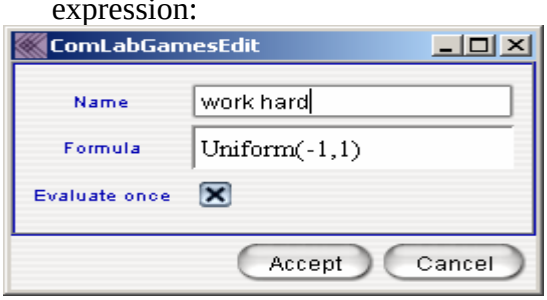


manager choice variable is defined as type "any". If you are unsure which type to use, you can select "any" and the program will select the appropriate type.



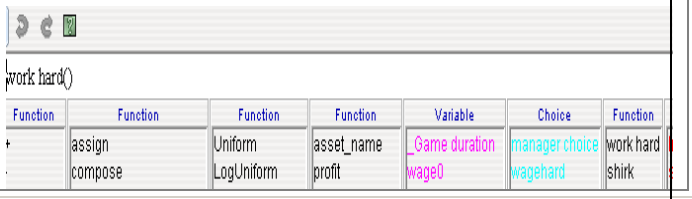
signal hard variable is defined as real type and it is updated with the function "work hard()":
The functions are defined in the following way:

a. drag and drop the function icon work hard. Function was renamed to "work hard".
b. Double click on work hard and write the expression:

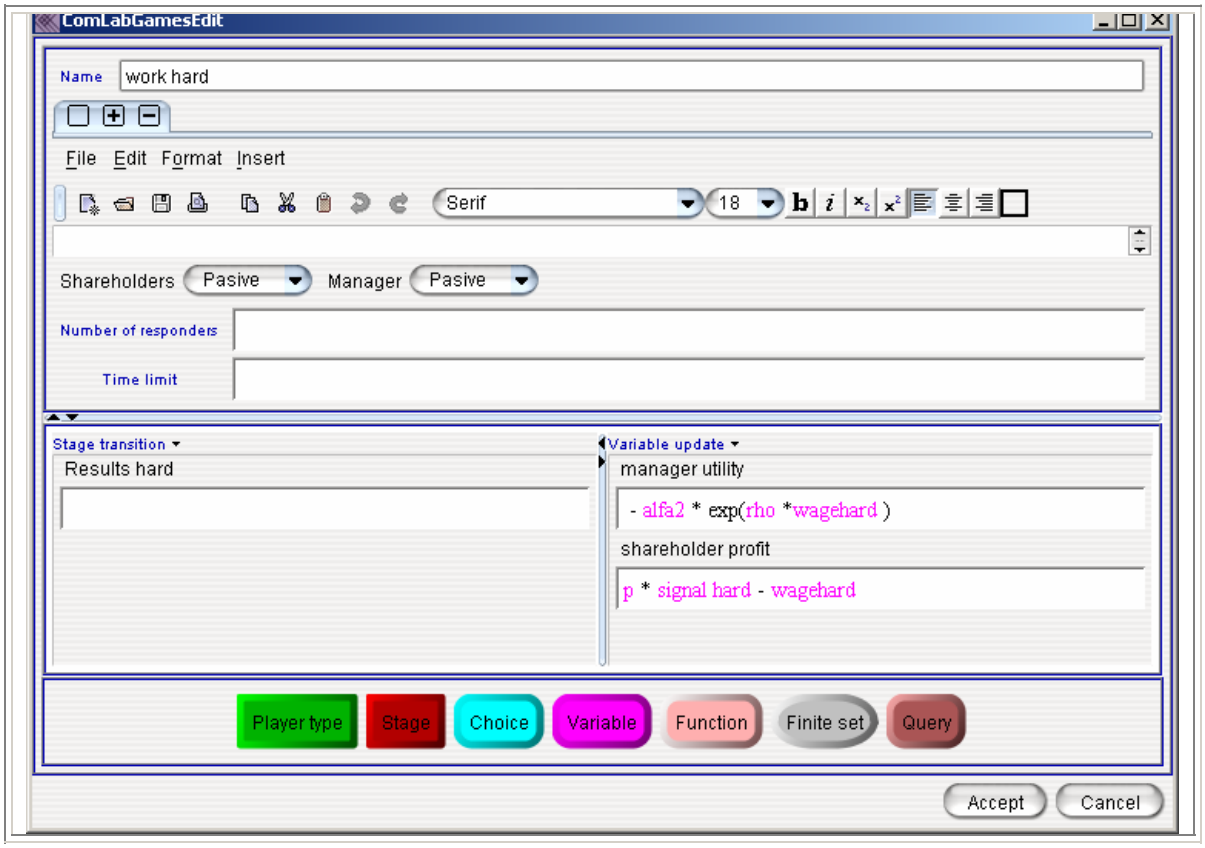


A number from a uniform distribution with the support -1 and 1 will be drawn. It will be evaluated once which means that each subject will get the same value.

c. Work hard() function can be seen in the Assistant editor:



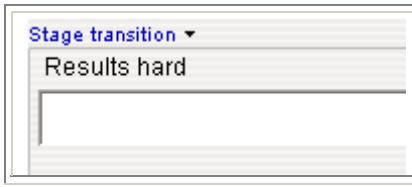
16. work hard **stage**



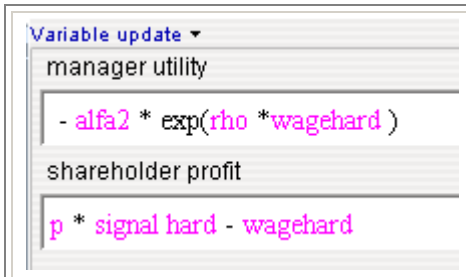
17. Player's task on work hard **stage**

Shareholders Pasive Manager Pasive	Both, shareholders and manager are "passive" which means that work hard will not be seen by neither player types. We can use these type of stages to prepare results (i.e. update variables). Subjects will not noticed any difference by having the stages introduced in this way.

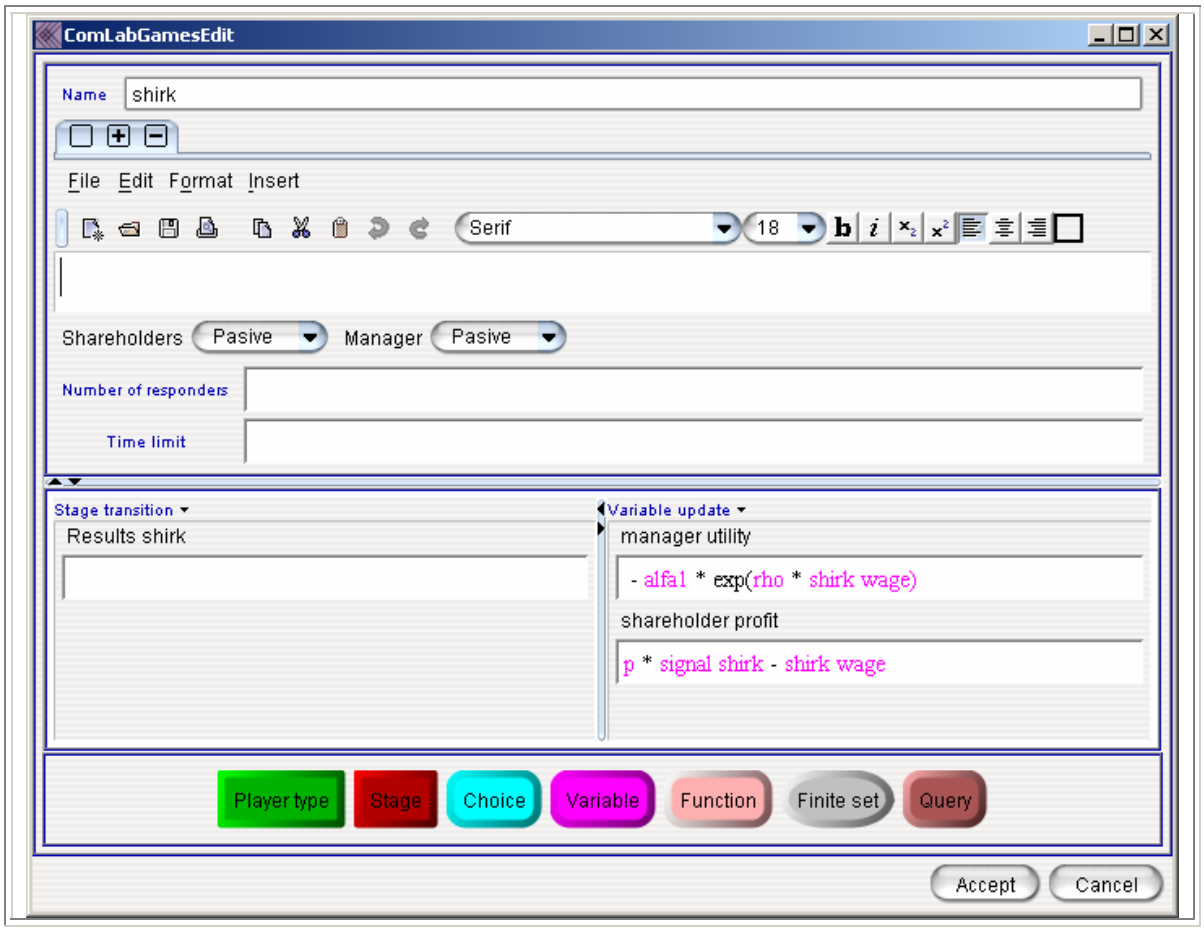
18. Stage transition on work hard **stage**

	Results hard stage is reached with probability 1.

19. Update of the variables on the work hard stage

	manager utility is defined as real number. The expression consists of the constants alpha2, rho and the variable wagehard that shareholders offered to manager and the manager chose to accept. shareholder profit is defined as real number.

20. shirk stage



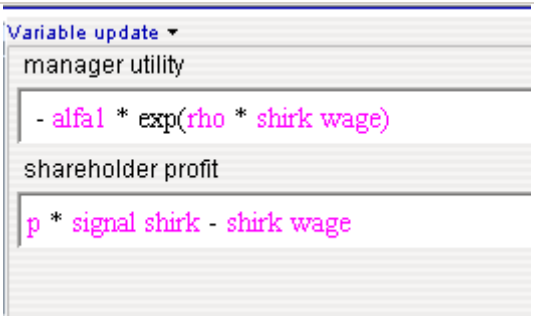
21. Player's task on shirk **stage**

Shareholders Pasive Manager Pasive	Both, shareholders and manager are "passive" which means that shirk will not be seen by neither player types. We can use these type of stages to prepare results (i.e. update variables). Subjects will not noticed any difference by having the stages introduced in this way.

22. Stage transition on shirk **stage**

Stage transition Results shirk	Results shirk stage is reached with probability 1.

23. Update of the variables on the shirk stage



manager utility is defined as real number. The expression consists of the constants alpha1, rho and the variable shirk wage that shareholders offered to manager and the manager chose to accept.

shareholder profit is defined as real number.

24. Result presentation for 

(i.e. similarly for all the other stages)

Note that the variables are inserted from the Insert menu.

Also 

are both active if we want them to see this information.