The JavaPairing User Manual

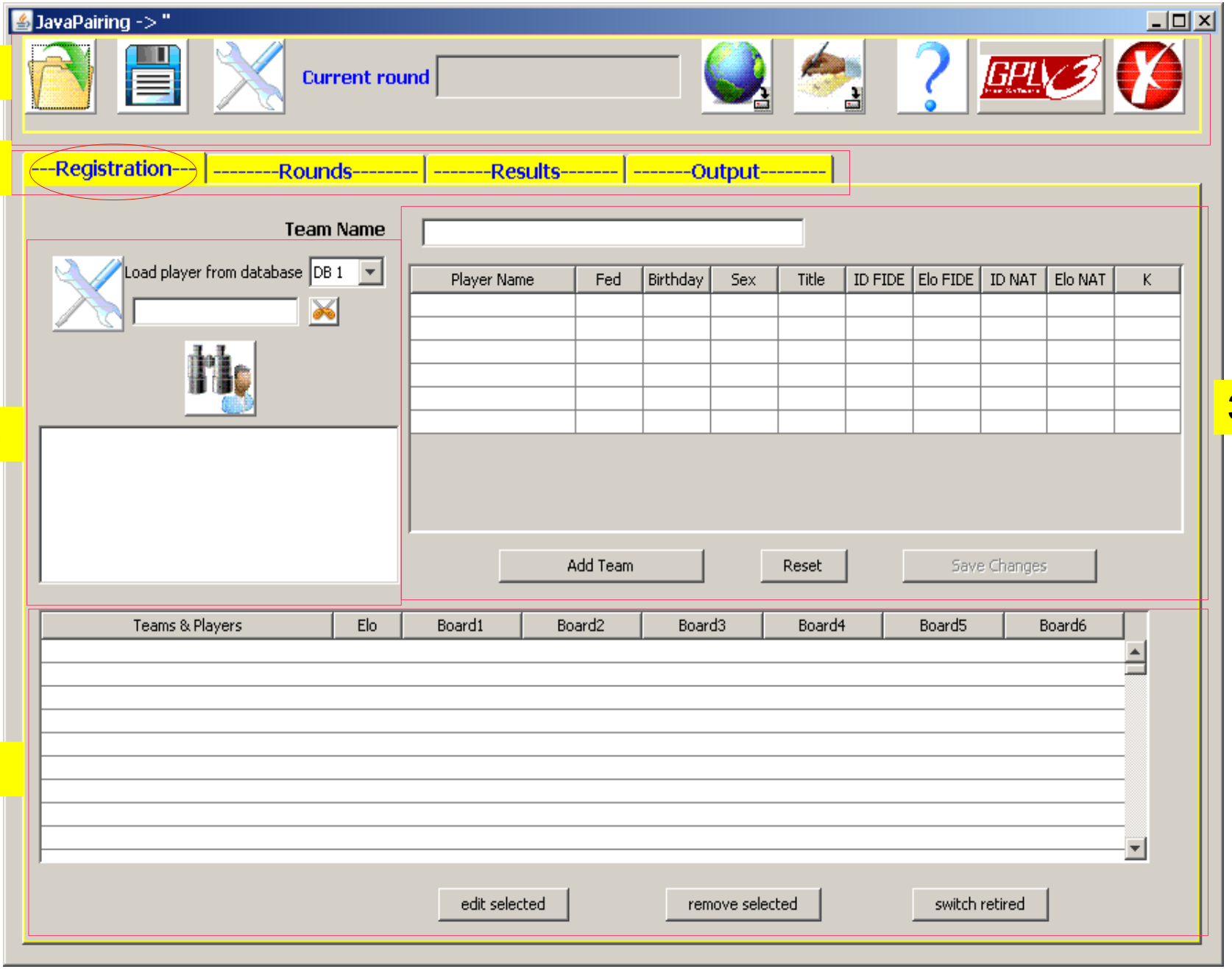
Summary

JavaPairing - rev. 2.6 – June 2013

- This is a tool to manage a chess tournament, team or individual (may set teams of one player! -:)
- I decided to start this work because of some unsatisfaction with commercial software, need to workaround some bugs, to overcome some unpleasant restrictions, to improve the user interface, to flow and speed up the work of the arbiter on the field.
- The program is pure Java and can run as is on any platform, including Windows, Linux and Mac (JRE 1.6 or more required). It is free and open source under the GPLv3 license. See the license file for details.
- Hope this may result in a cooperative international work with the goal to obtain FIDE endorsment.
- The program is now stable enough to be used in real tournaments.

Index to pages

The Main Window



1 **The toolbar** with 'Open tournament', 'Save tournament', 'Setup tournament', 'round counter', 'web site gen.', 'report gen.', 'help', 'license', 'Quit' buttons.

2 **Page switcher** ('Registration' page is active. For other pages see next). Each task of the arbiter is easily assisted in those specialized pages.

3 **Team name and detail Player data** for Team to add or edit. No field is mandatory except obviously the name of the player(s).

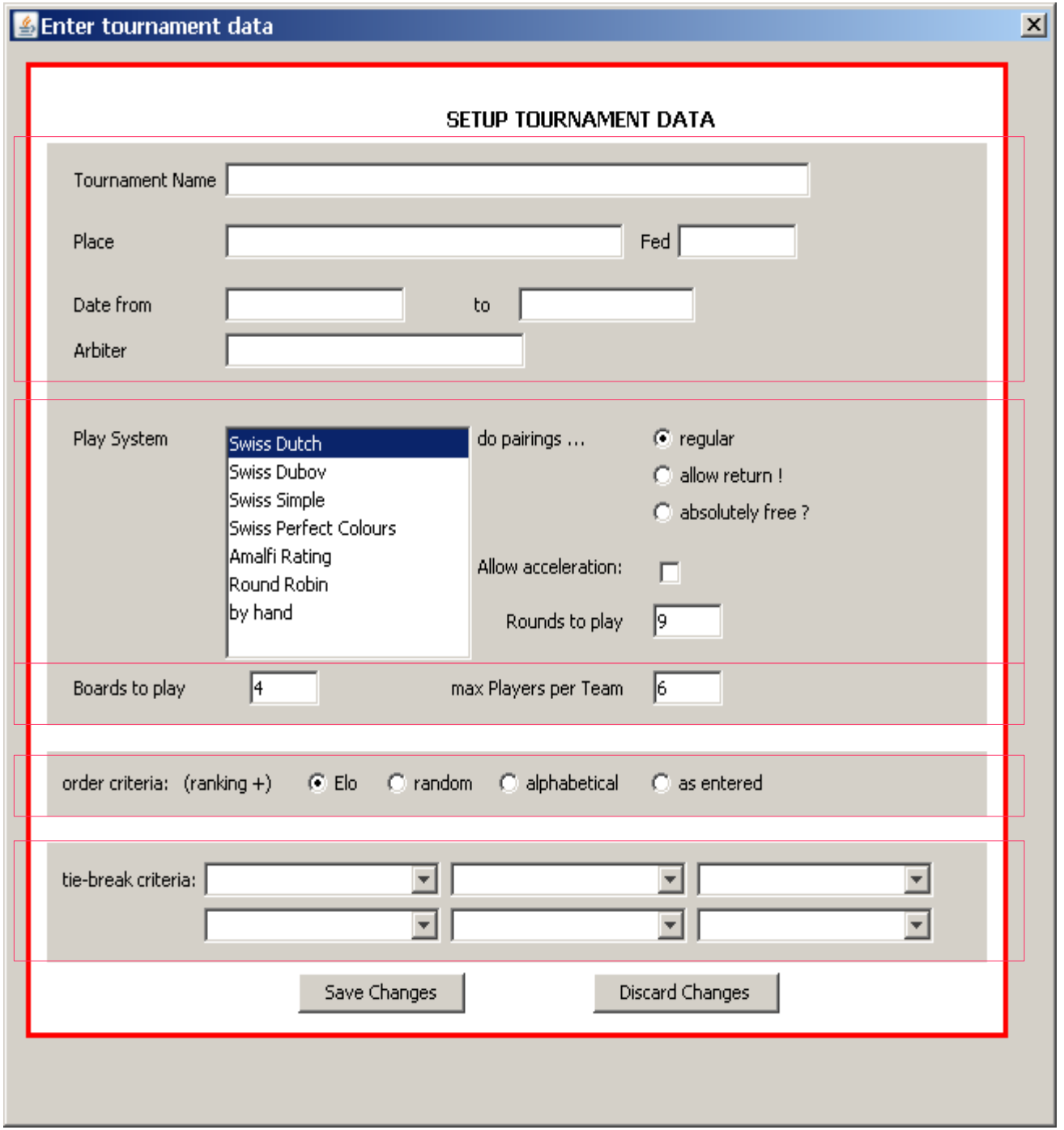
4 **Area to setup, search and load Players** from commonly distributed text databases. Up to 5 filters should be easily applied simultaneously, after that you can drag&drop (or double-click) players from the list object into the grid. Filters for FIDE, italian and german federations are included in the distribution.

5 **Archive of all Teams** (max 500) **and/or Players** entered, listed in the order they are entered. This grid is set read-only. Each row stores: the name of the Team, the mean Elo (of the holders only), and 1...N boards. Each board cell contains all player data packed into one string with fields delimited by ';'. This way all data is kept in memory, no temp disc file is used! **You may sort the grid by clicking on the column header or by using the right button of the mouse to fine tune the list until the first round is built.** Consider also the options in the Setup page.

n.b.:

- Grids are dynamic objects. Click and drag to resize columns.
- Late entrant may be safely added at any time, from the database too, as well as mistakes may be corrected. Anyway after the first round you cannot remove items
- Players may be switched 'retired', i.e. set 'unavailable' to be paired, any time.

The Tournament Setup Window



General data

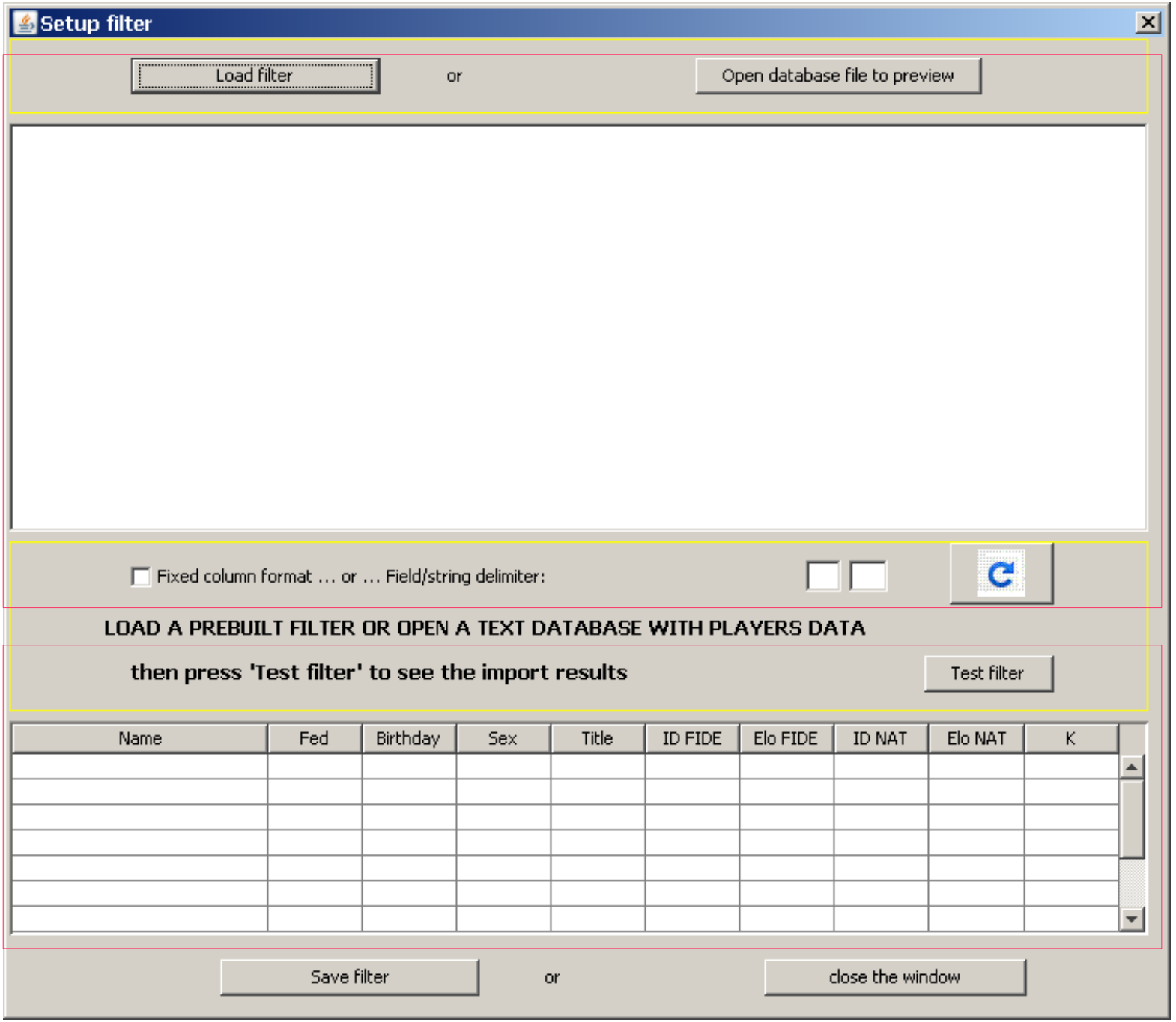
how to pair;
max 16 rounds

# of boards ≤ 10;
Set 1 for individual

how to order

tie-break selection

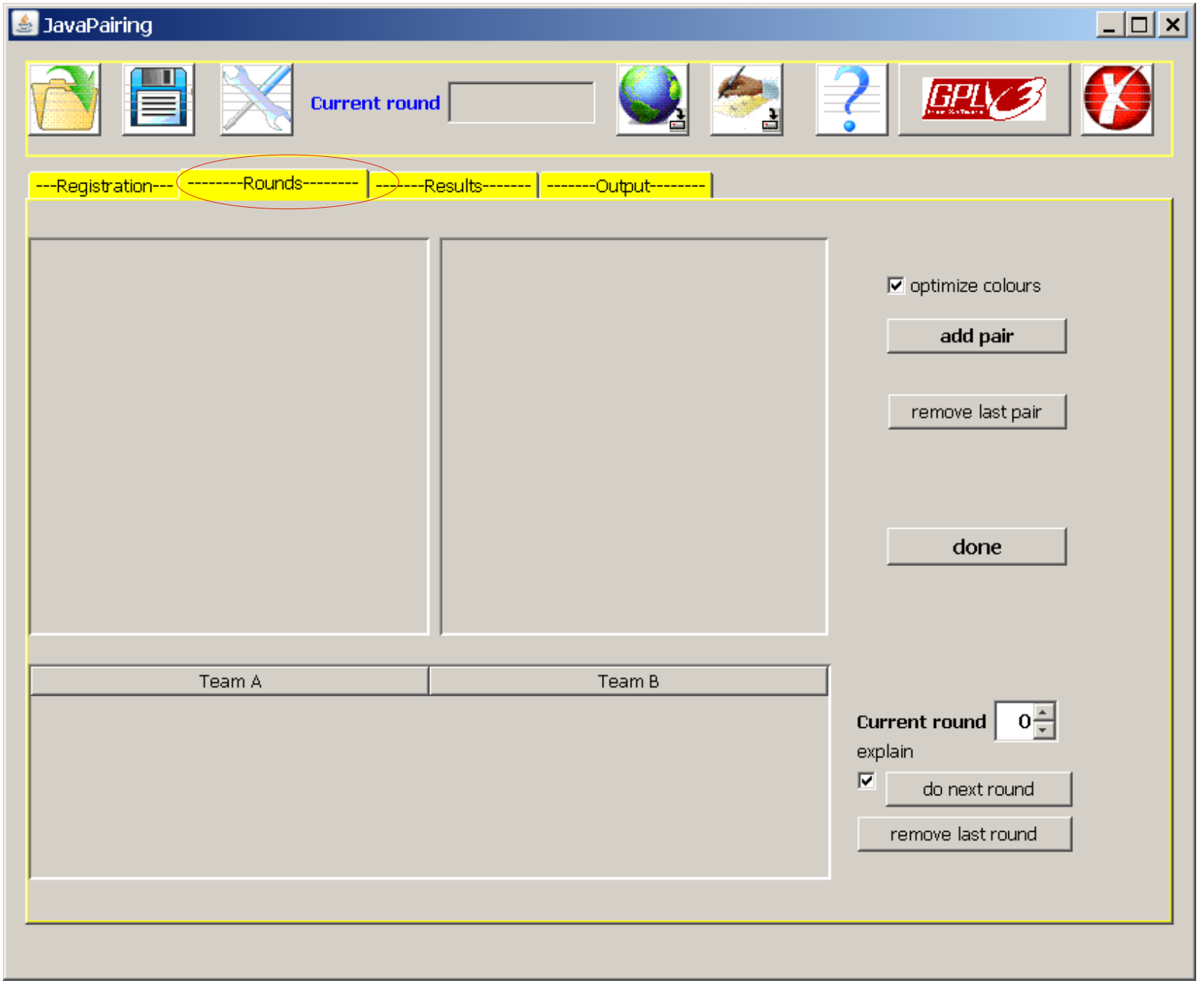
The load/create filter Window



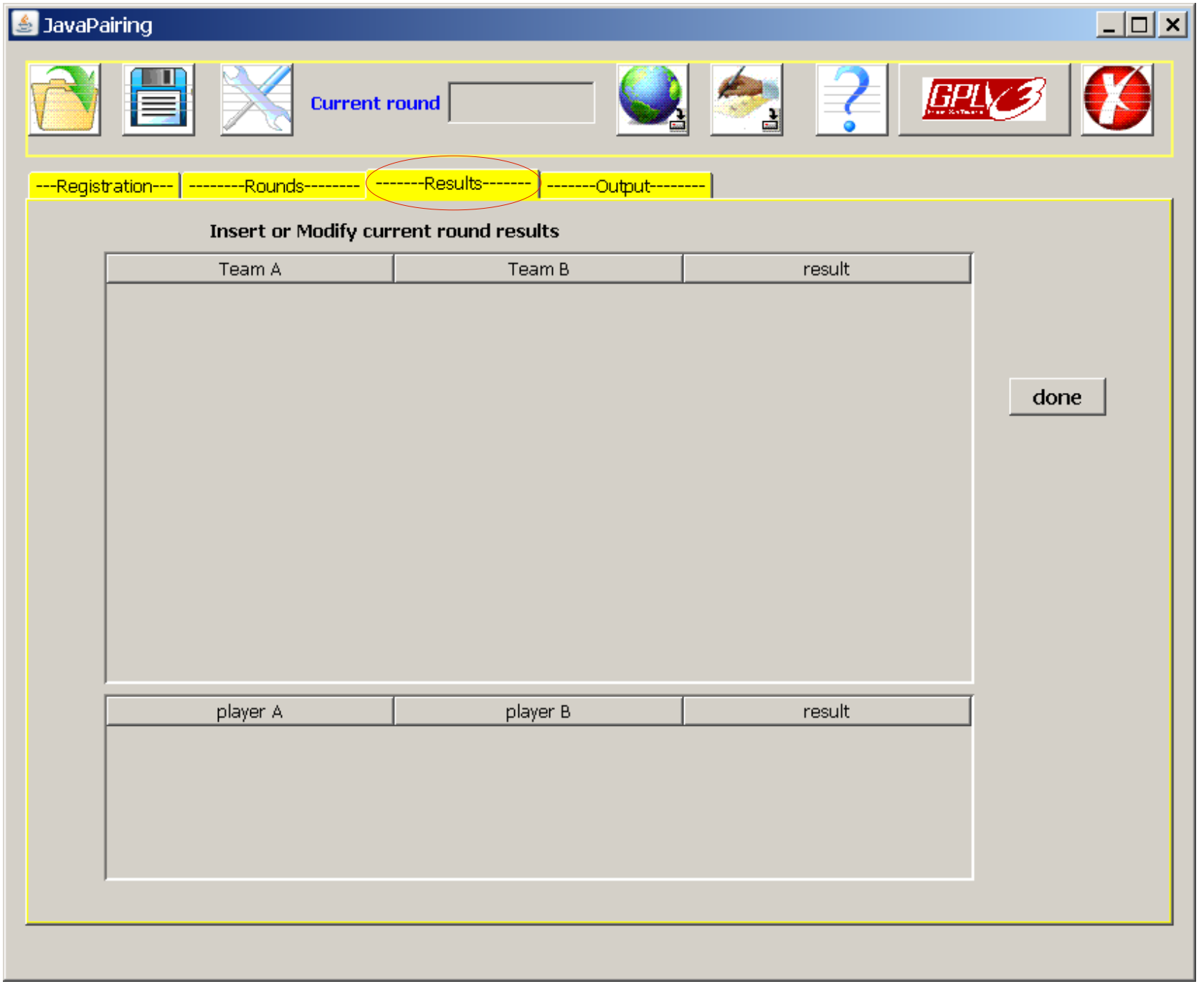
Preview data structure

Verify import success

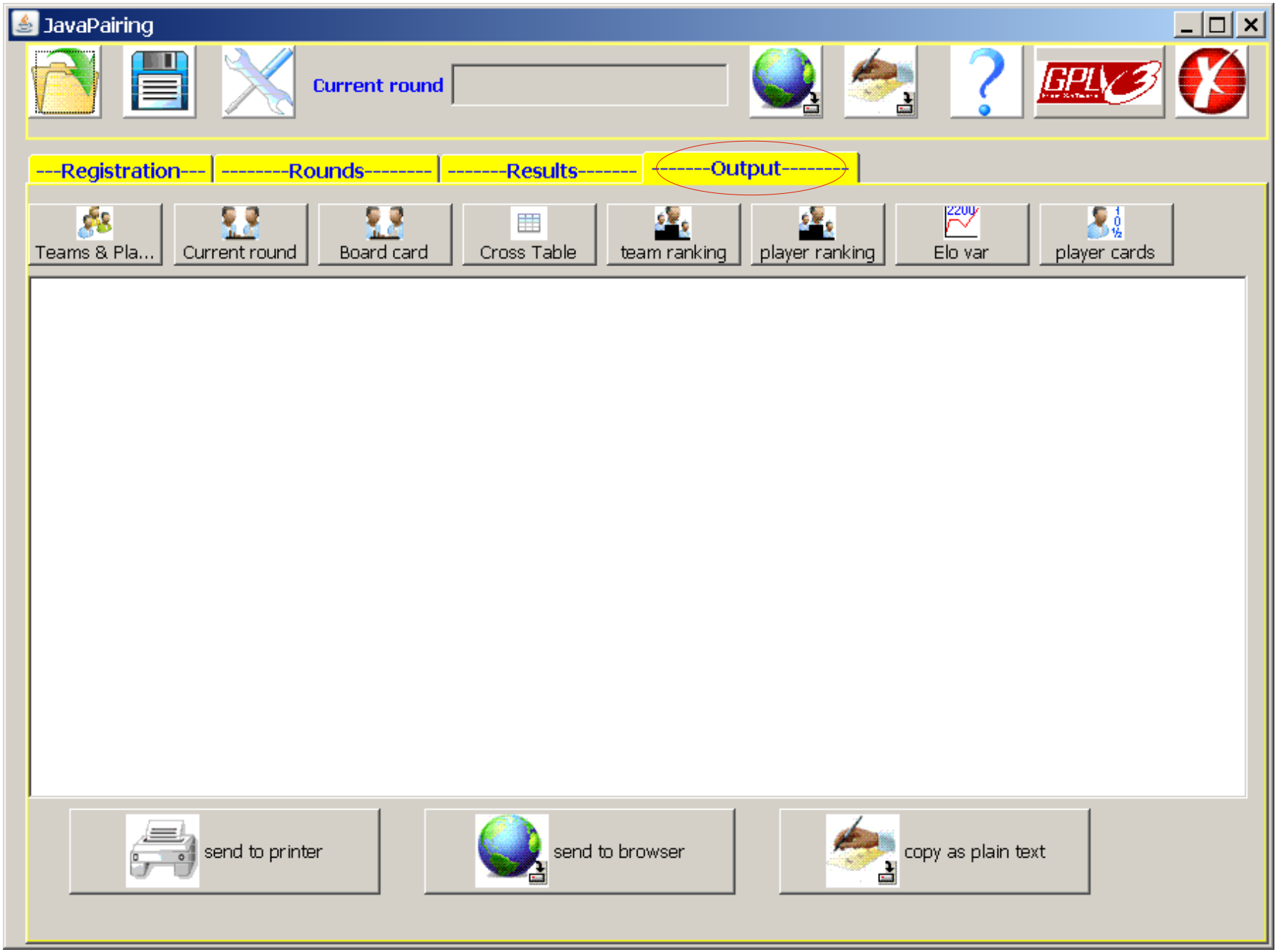
The rounds page



The results page



The output page



A sample tournament administration step by step

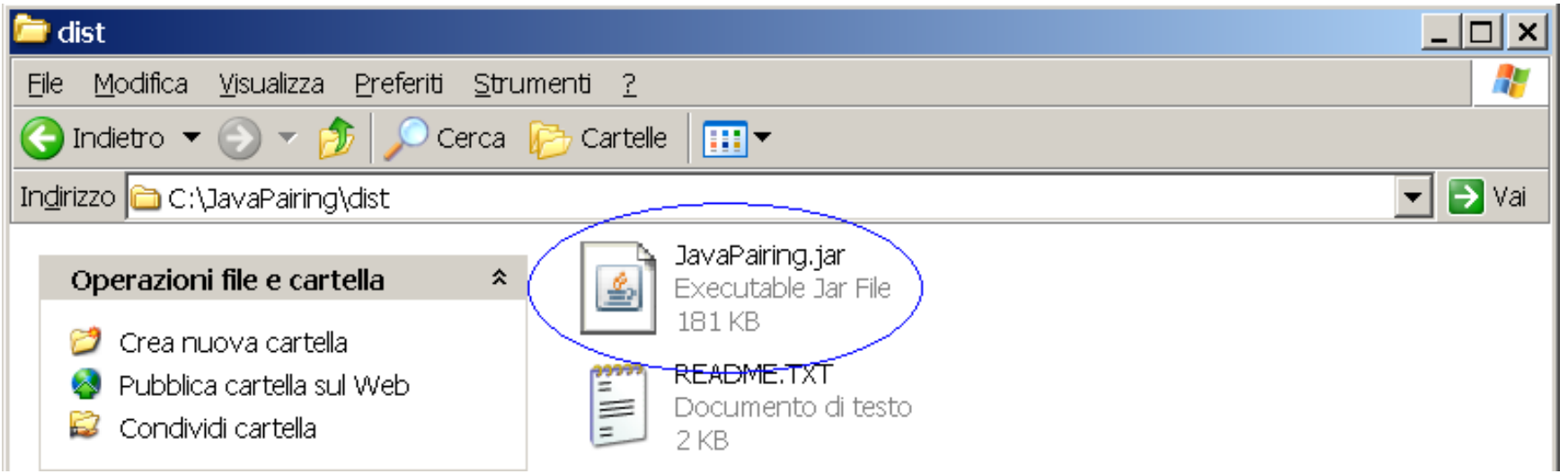
- Have you downloaded the latest official player database from FIDE or from your national federation? There may be two databases: 'Elo' and 'Elo rapid' rating. (advanced users: you may maintain player databases by yourself. The program can read plain text files, either fixed format or comma separated value. Those formats may be generated i.e. with Excel or OpenOffice by their export capabilities).

- Have you decided how many rounds to play? Want a regular tournament (i.e. each pair can be generated only once), a return tournament (i.e. each pair can be generated twice), or an absolutely free pairing tournament (this is an experimental option and I don't know the usefulness, the program simply does not check for previous pairing). During the tournament you may safely switch between options. I suggest you to start with 'regular', and after completing the required rounds switch to 'return' to allow return match (with alternated colours if optimization is requested).

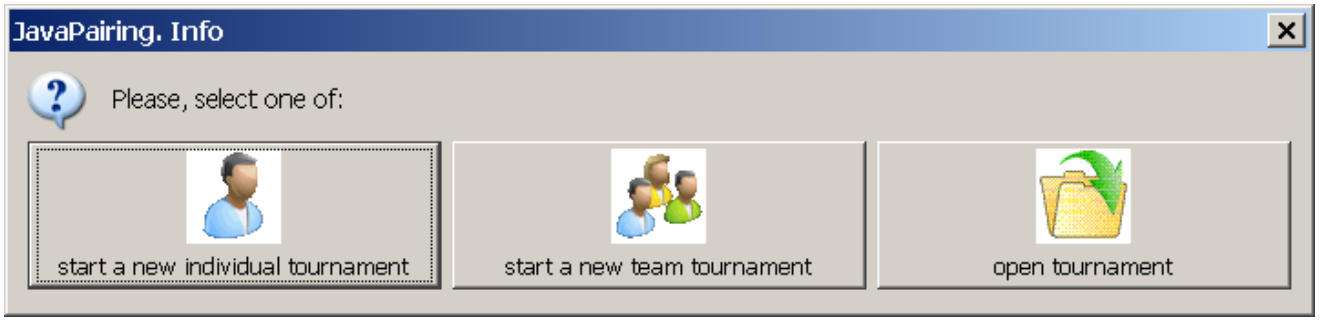
- Have you decided which system to play with? The possible choices are 'swiss Dutch', 'swiss Dubov', 'swiss Simple', 'swiss Perfect Colours', 'Amalfi Rating', 'Round Robin' and 'by hand' i.e. you have to build pairs by yourself! In the last case, the program helps you much ranking the Teams and highlighting previous opponents. The program helps to optimize colour selection too. Remember it's your responsibility of arbiter to make correct pairs! You can find state of the art documentation on main pairing systems in some sub folders of this distribution.

... well, let's start our first tournament! I assume a standard installation on Windows platform i.e. the distribution has been unzipped in 'C:\JavaPairing'. Small variations should be introduced for other platforms.

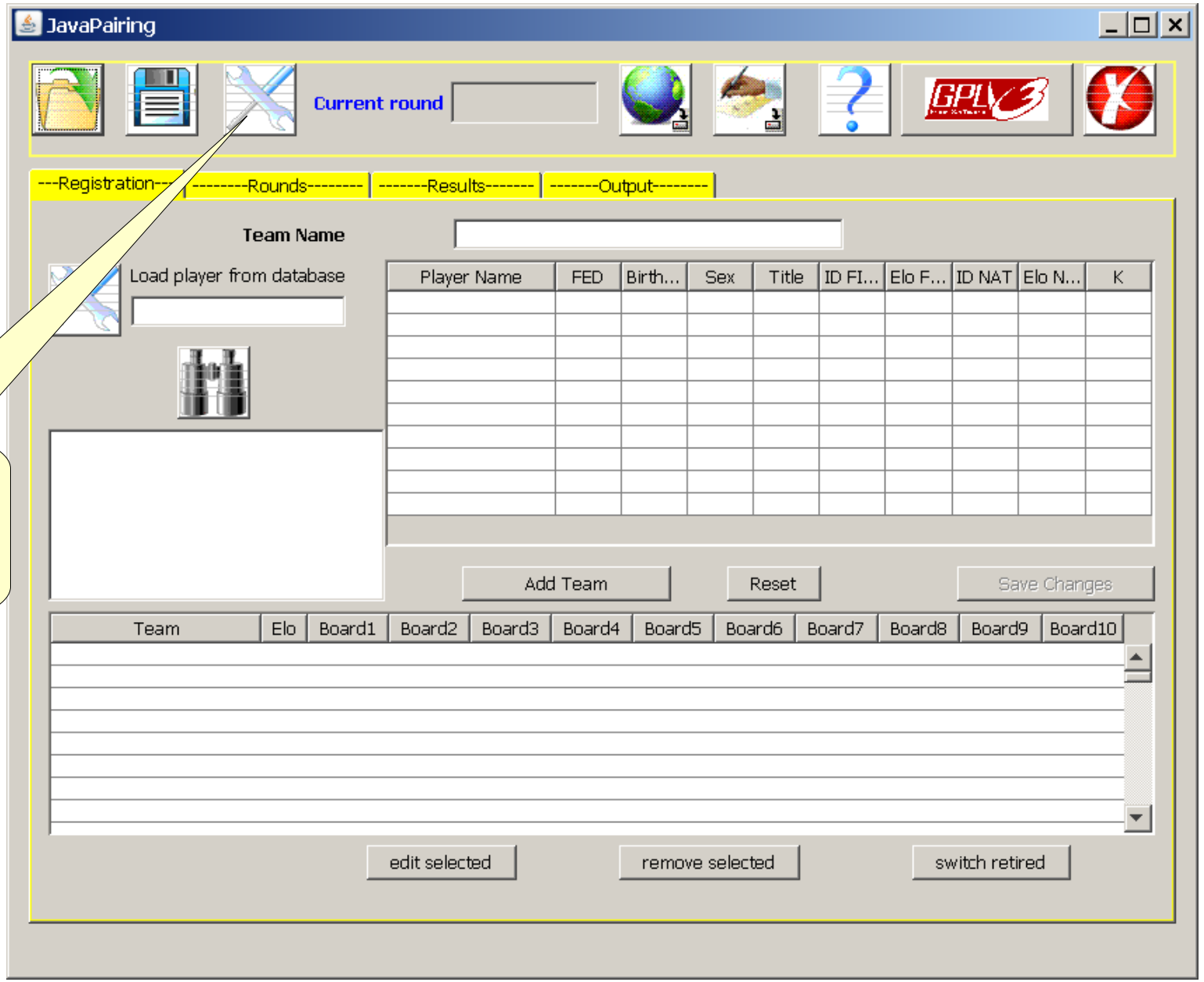
So, browse to the installation folder, go to the 'dist' sub folder and double click on the program icon



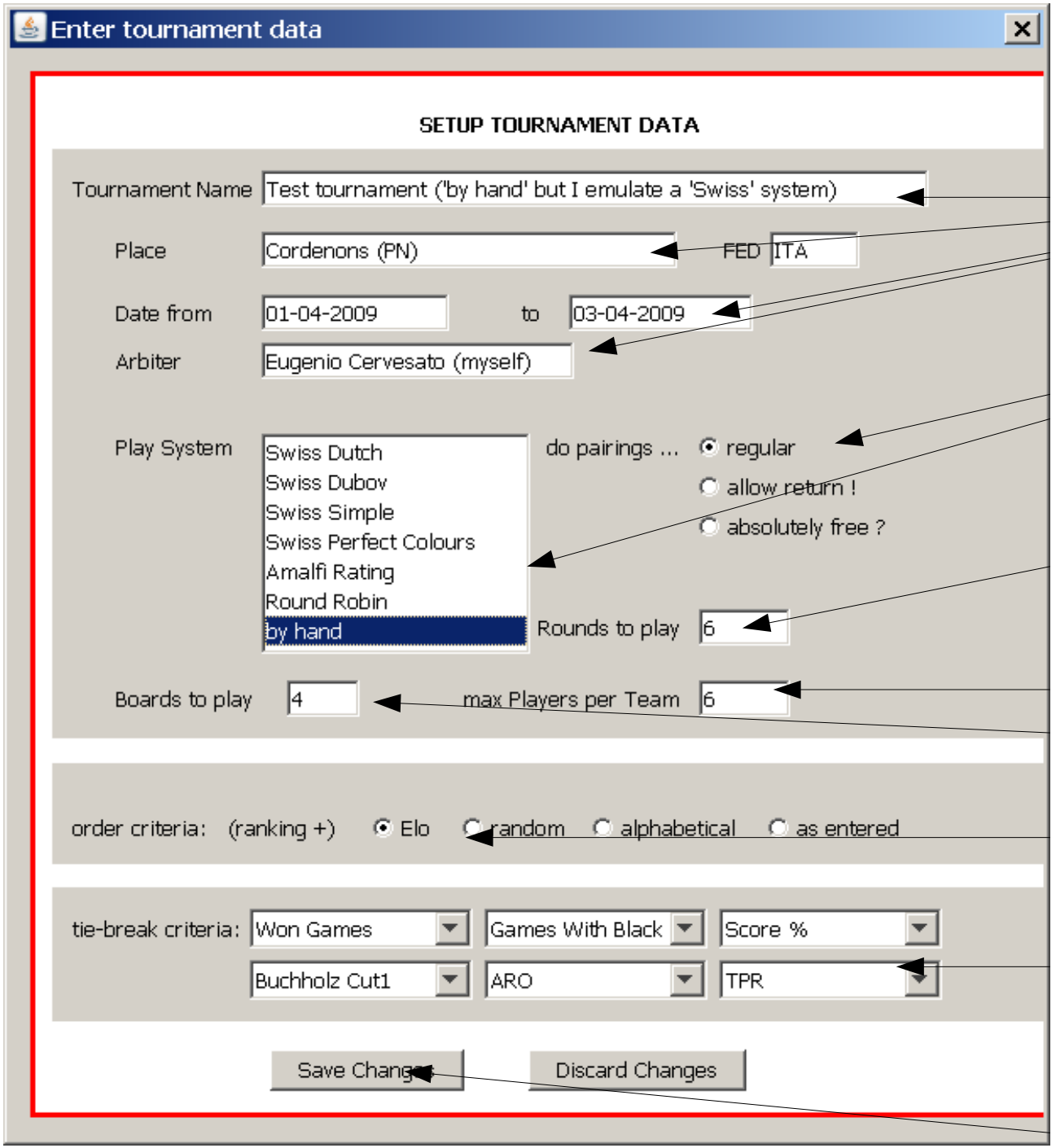
If all is fine with the installation (remember JRE 1.6 or more is required), the assistant opens ...



... to follow the sample, select the second button ...



1.st click the 'Setup tournament' icon!



General data. A regular Swiss tournament

play system

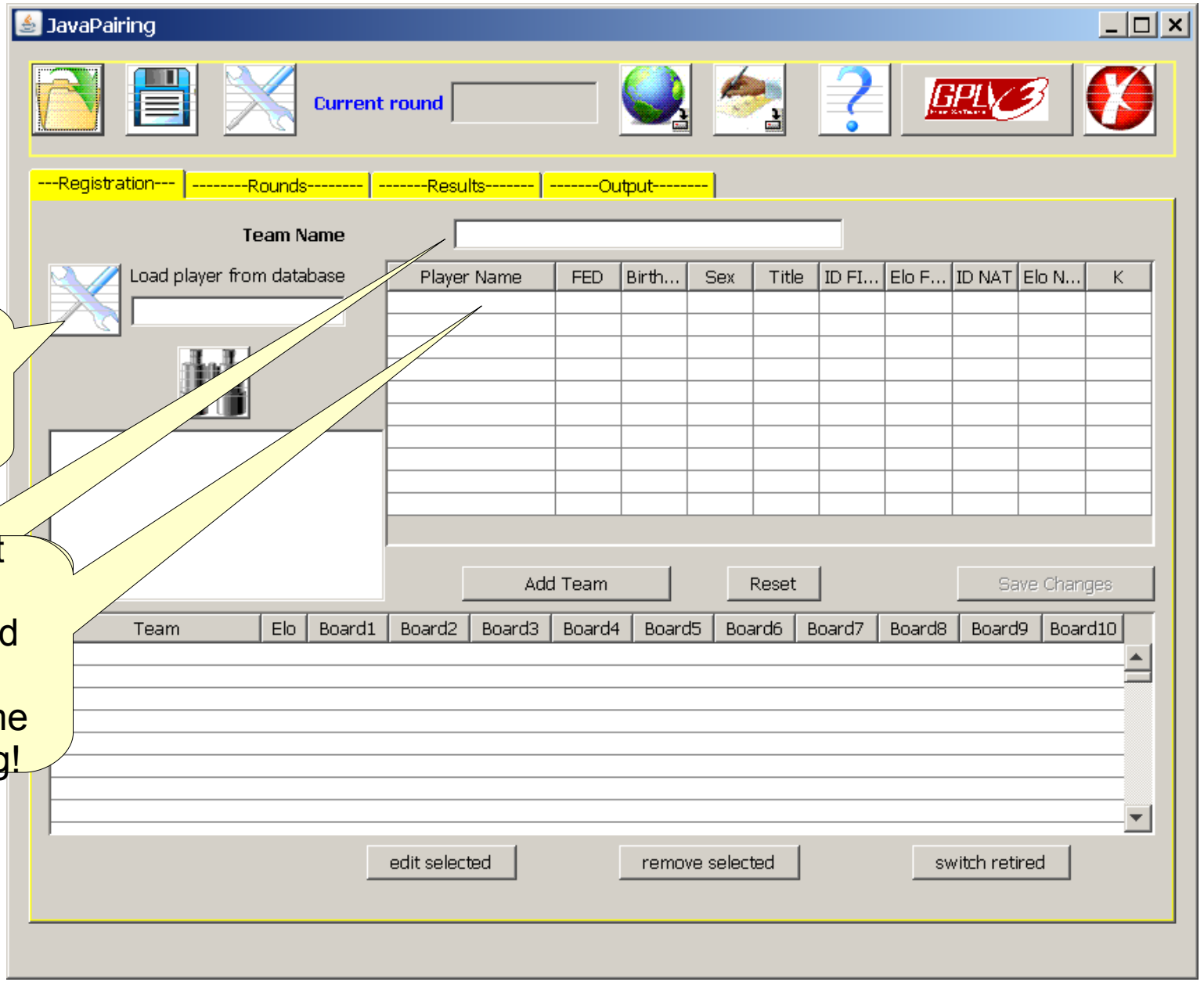
6 rounds to play

boards. Ok. default
(Teams of 4 holders and max 2 reserves)

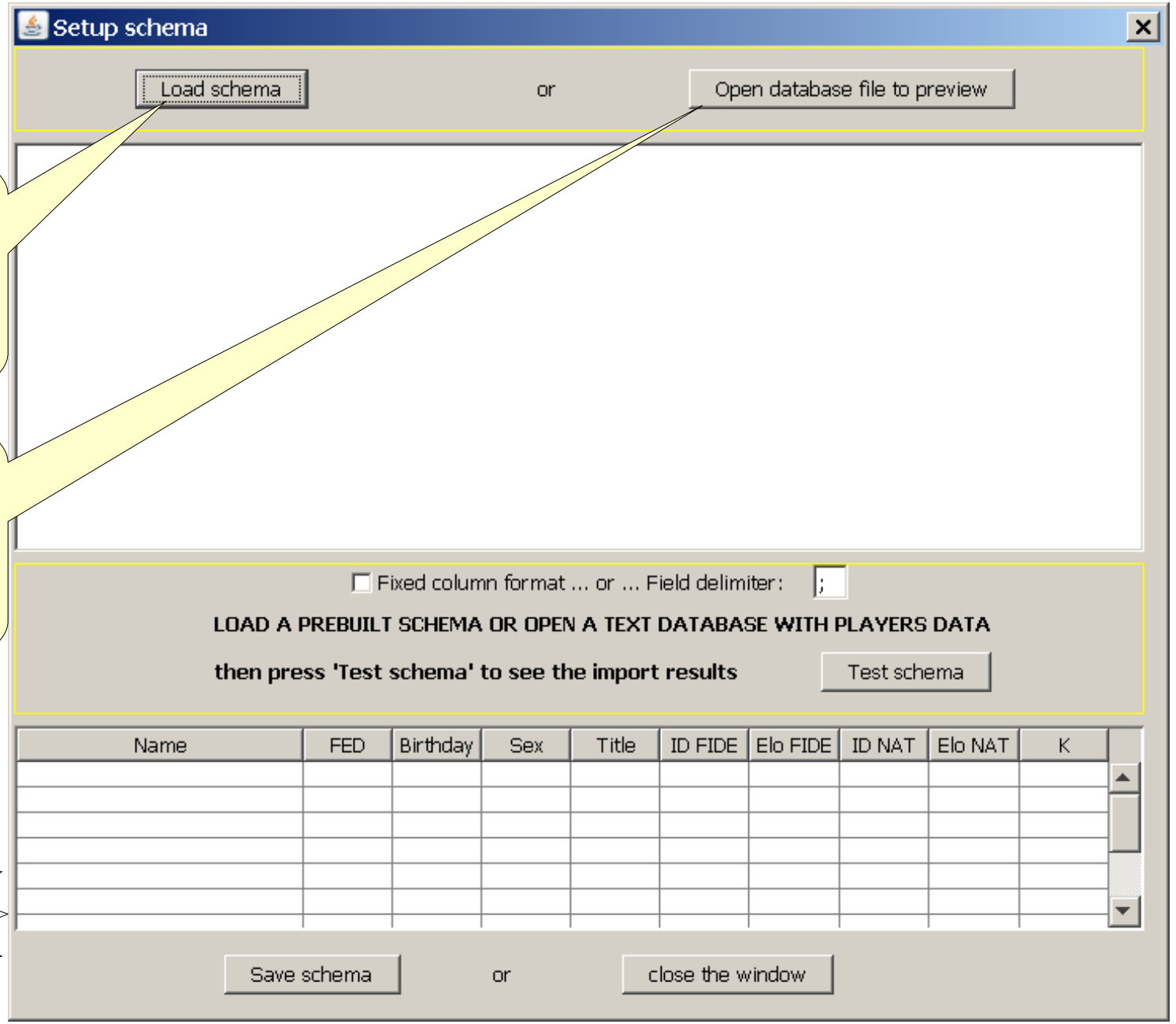
order by Elo 1.st round, by ranking + Elo next rounds

tie-break selection; max 6 criteria at a time

save and exit the window


JavaPairing
Current round
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Team Name
Load player from database
Player Name
FED
Birth...
Sex
Title
ID FI...
Elo F...
ID NAT
Elo N...
K
Add Team
Reset
Save Changes
Team
Elo
Board1
Board2
Board3
Board4
Board5
Board6
Board7
Board8
Board9
Board10
edit selected
remove selected
switch retired
2.nd go to the 'load/create filter' page! ... very easy!
... or start typing in Teams and Players data! ... time consuming!

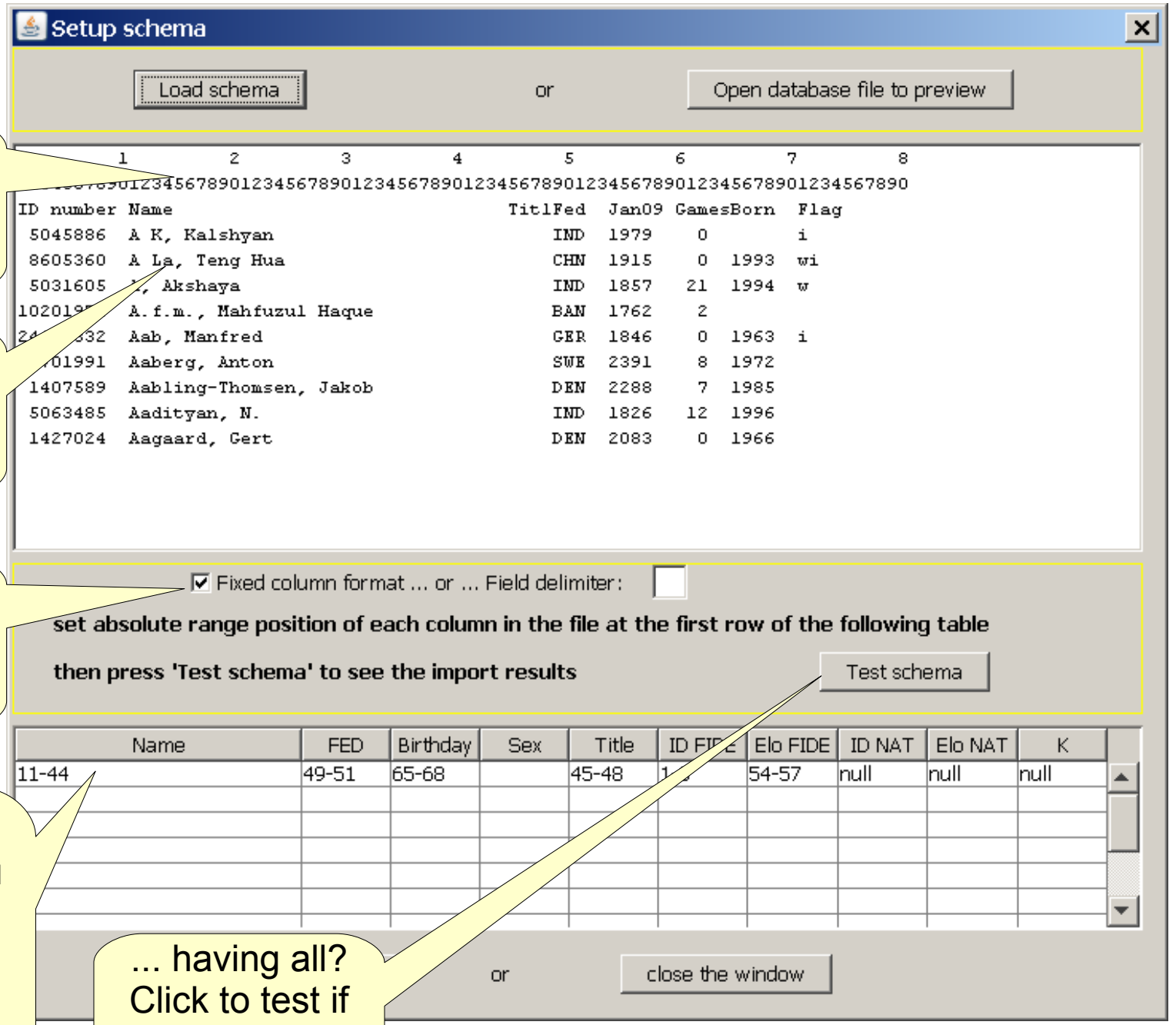
a) in case of fixed format ...



Load prebuilt filter file: 'C:\JavaPairing\DATABASES\FIDE.scm' ...

... or open a text database: 'C:\JavaPairing\DATABASES\jan09frl.txt'

The two choices are not exclusive!



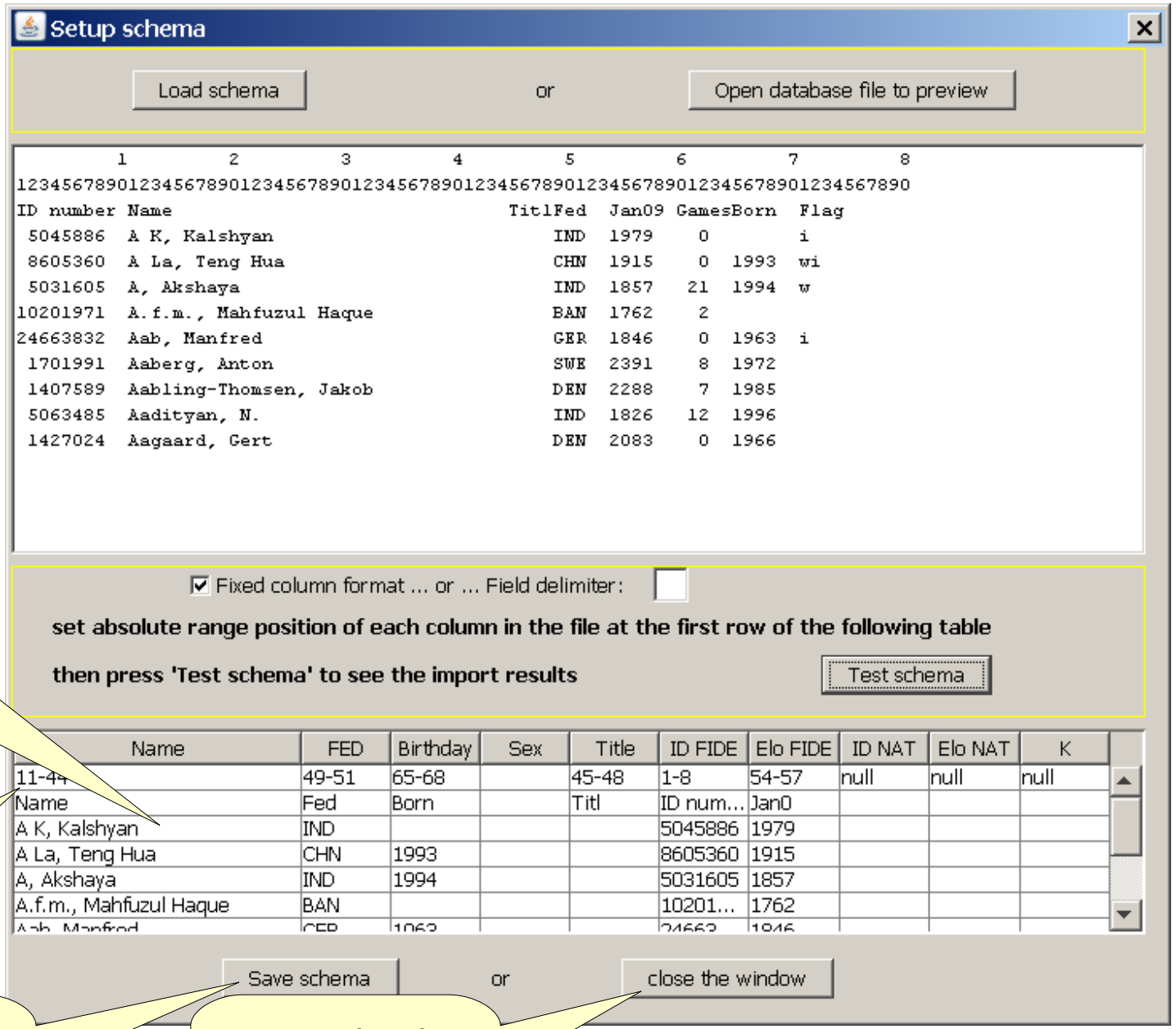
This is a fixed length file, so a ruler is offered ...

Preview of Players data follows ...

This is a fixed column format, no delimiter ...

At the first row of the grid / you have / have to set / the known column start-end values ...

... having all? Click to test if the filter works ...



... if the filter works you have columns with corresponding good data ...

... if not, adjust column header and repeat ...

... save filter 1.st time if needed ...

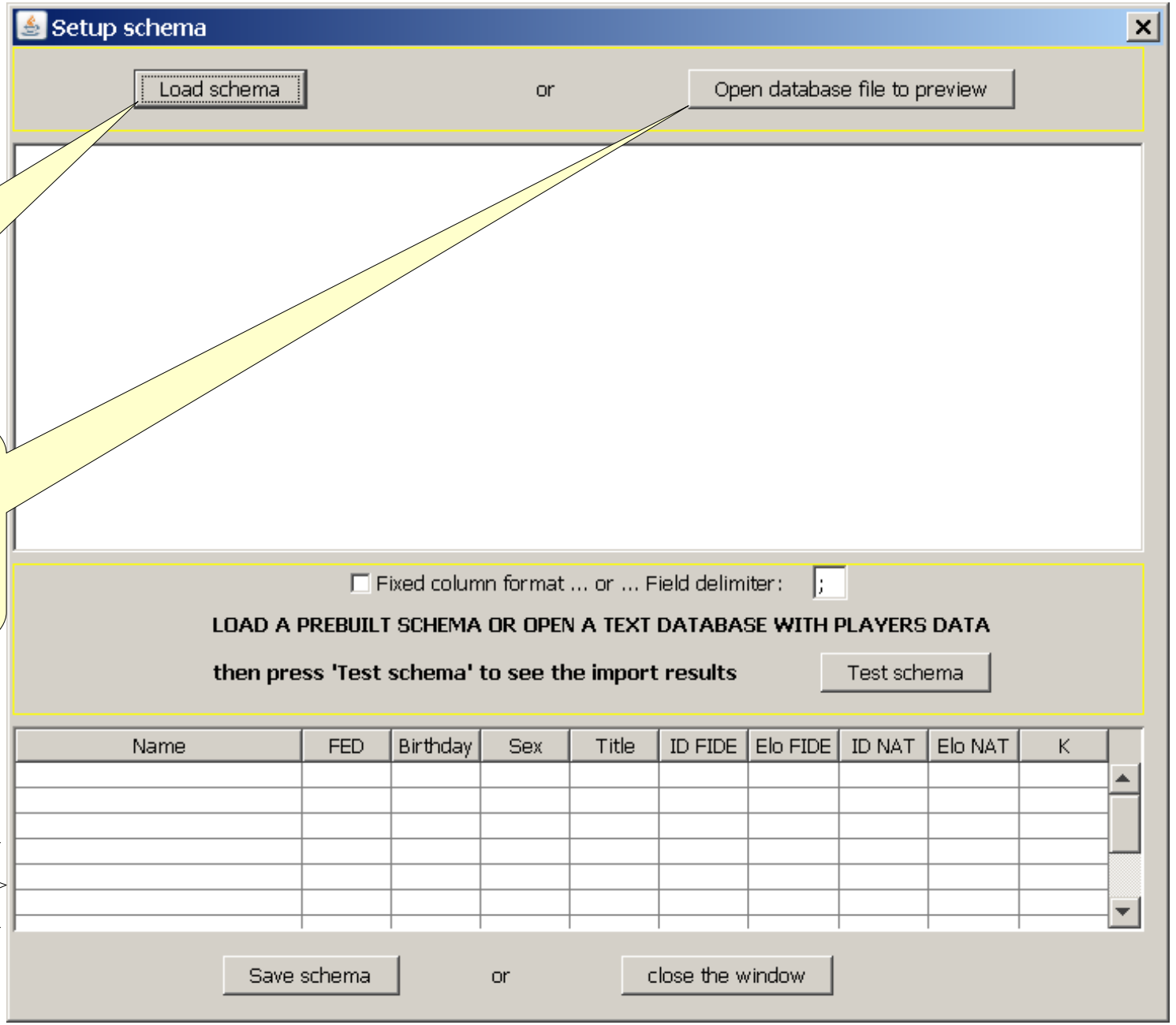
... or simply close the window ...

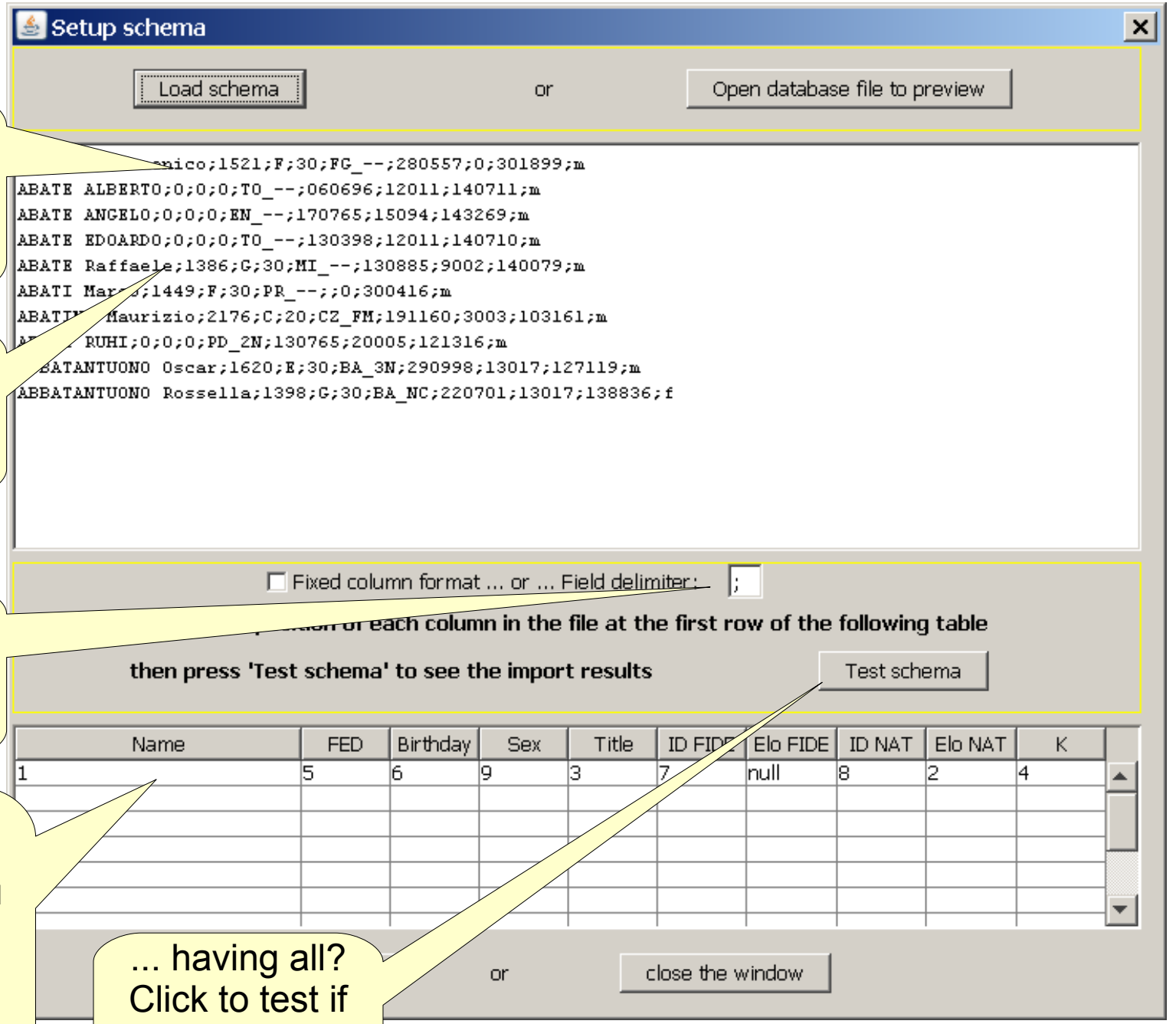
b) in case of variable format ...

Load prebuilt filter file: 'C:\JavaPairing\ DATABASES\ FSI-csv.scm' ..

... or open a text database: 'C:\JavaPairing\ DATABASES\ vegadb.csv'

The two choices are not exclusive!





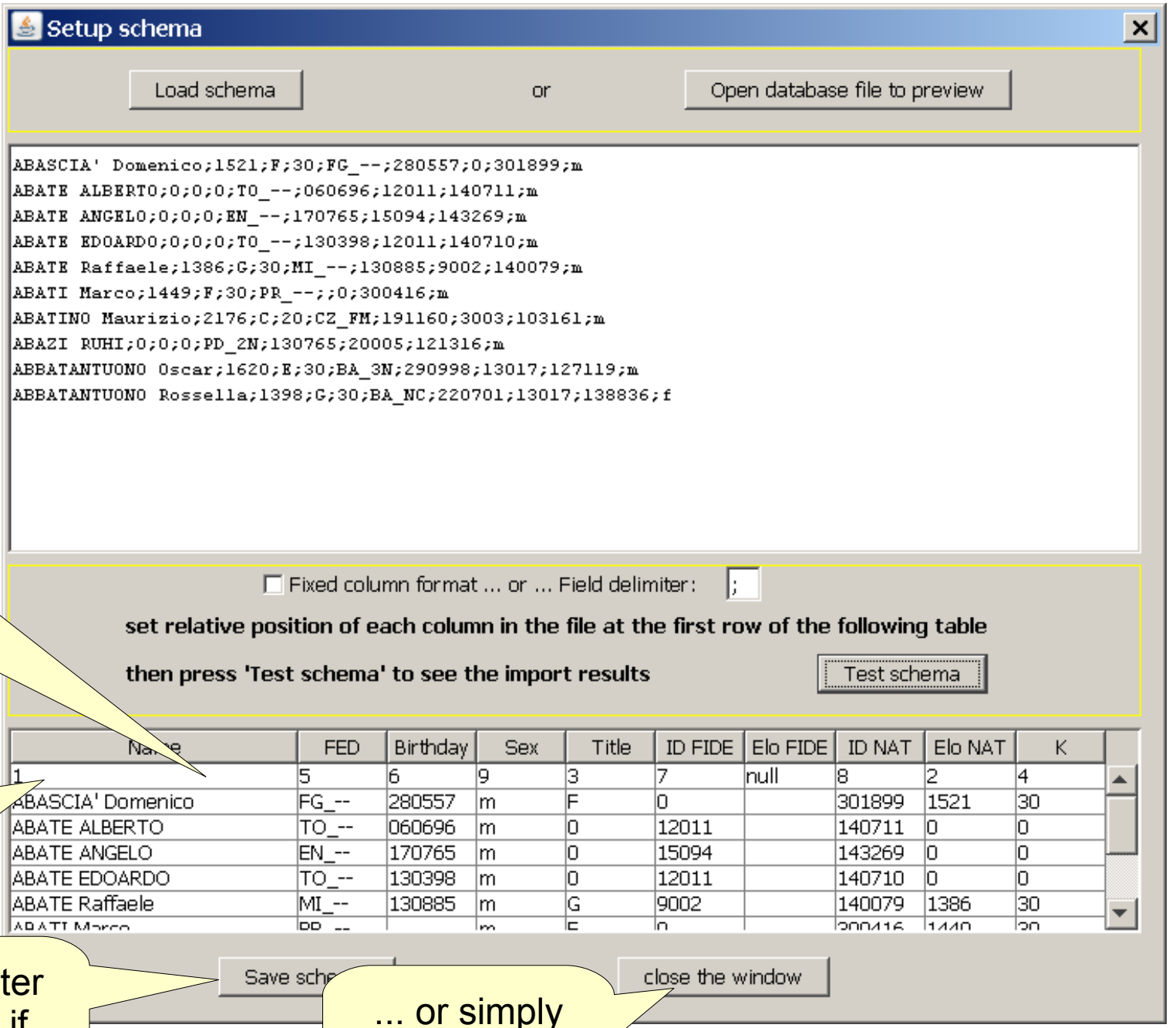
This is a variable length file, so a ruler has no sense ...

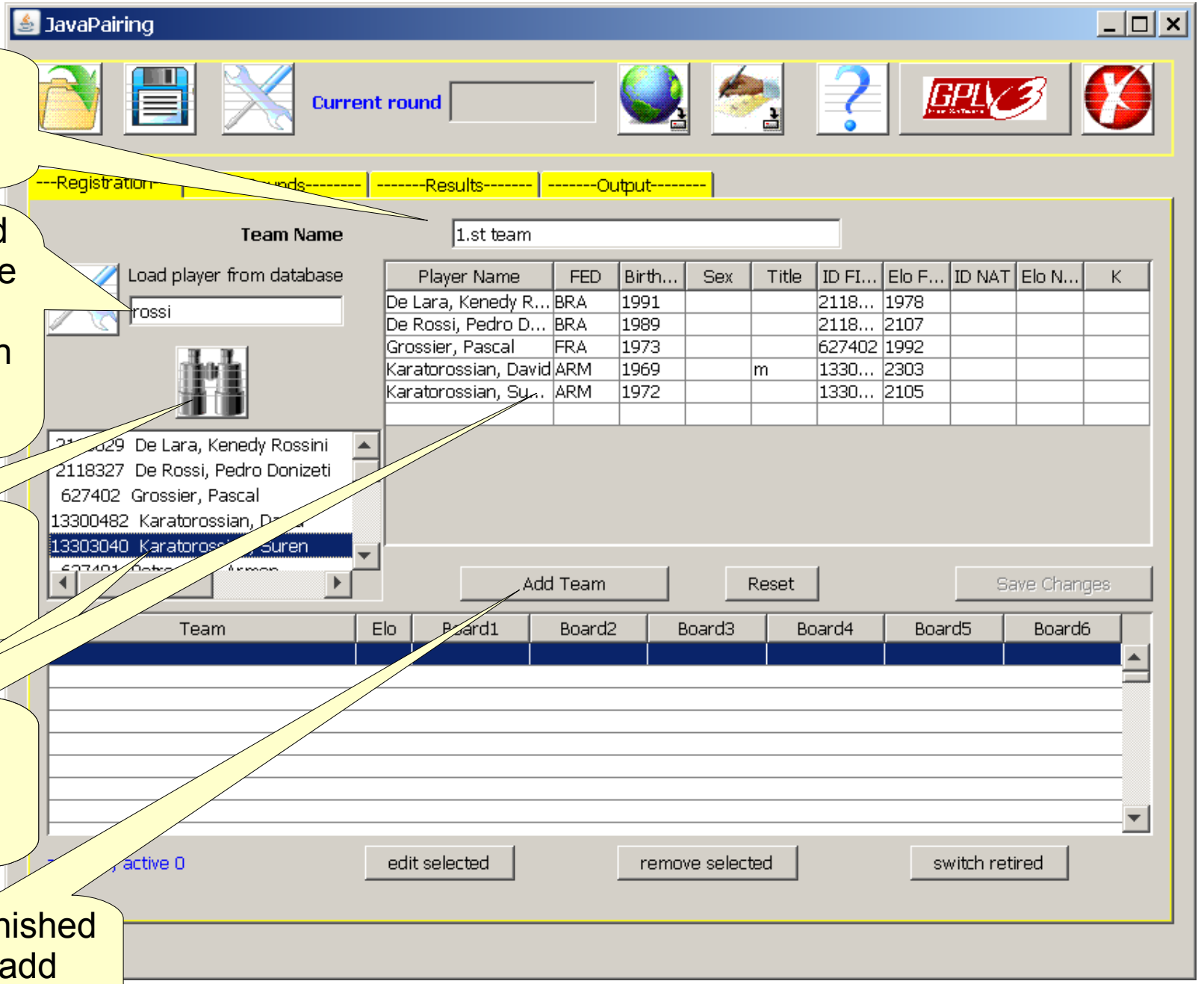
Preview of Players data ...

This is a delimited column format ...

At the first row of the grid / you have / have to set / the known column order ...

... having all? Click to test if the filter works ...


Setup schema
Load schema
or
Open database file to preview
ABASCIA' Domenico;1521;F;30;FG_--;280557;0;301899;m
ABATE ALBERTO;0;0;0;TO_--;060696;12011;140711;m
ABATE ANGELO;0;0;0;EN_--;170765;15094;143269;m
ABATE EDOARDO;0;0;0;TO_--;130398;12011;140710;m
ABATE Raffaele;1386;G;30;MI_--;130885;9002;140079;m
ABATI Marco;1449;F;30;PR_--;;0;300416;m
ABATINO Maurizio;2176;C;20;CZ_FM;191160;3003;103161;m
ABAZI RUHI;0;0;0;PD_ZN;130765;20005;121316;m
ABBATANTUONO Oscar;1620;E;30;BA_3N;290998;13017;127119;m
ABBATANTUONO Rossella;1398;G;30;BA_NC;220701;13017;138836;f
Fixed column format ... or ... Field delimiter:
;
set relative position of each column in the file at the first row of the following table
then press 'Test schema' to see the import results
Test schema
Name
FED
Birthday
Sex
Title
ID FIDE
Elo FIDE
ID NAT
Elo NAT
K
1
5
6
9
3
7
null
8
2
4
ABASCIA' Domenico
FG_--
280557
m
F
0
301899
1521
30
ABATE ALBERTO
TO_--
060696
m
0
12011
140711
0
0
ABATE ANGELO
EN_--
170765
m
0
15094
143269
0
0
ABATE EDOARDO
TO_--
130398
m
0
12011
140710
0
0
ABATE Raffaele
MI_--
130885
m
G
9002
140079
1386
30
close the window
... if the filter works you have columns with corresponding good data ...
... if not, adjust column header and repeat ...
... save filter 1.st time if needed ...
... or simply close the window ...



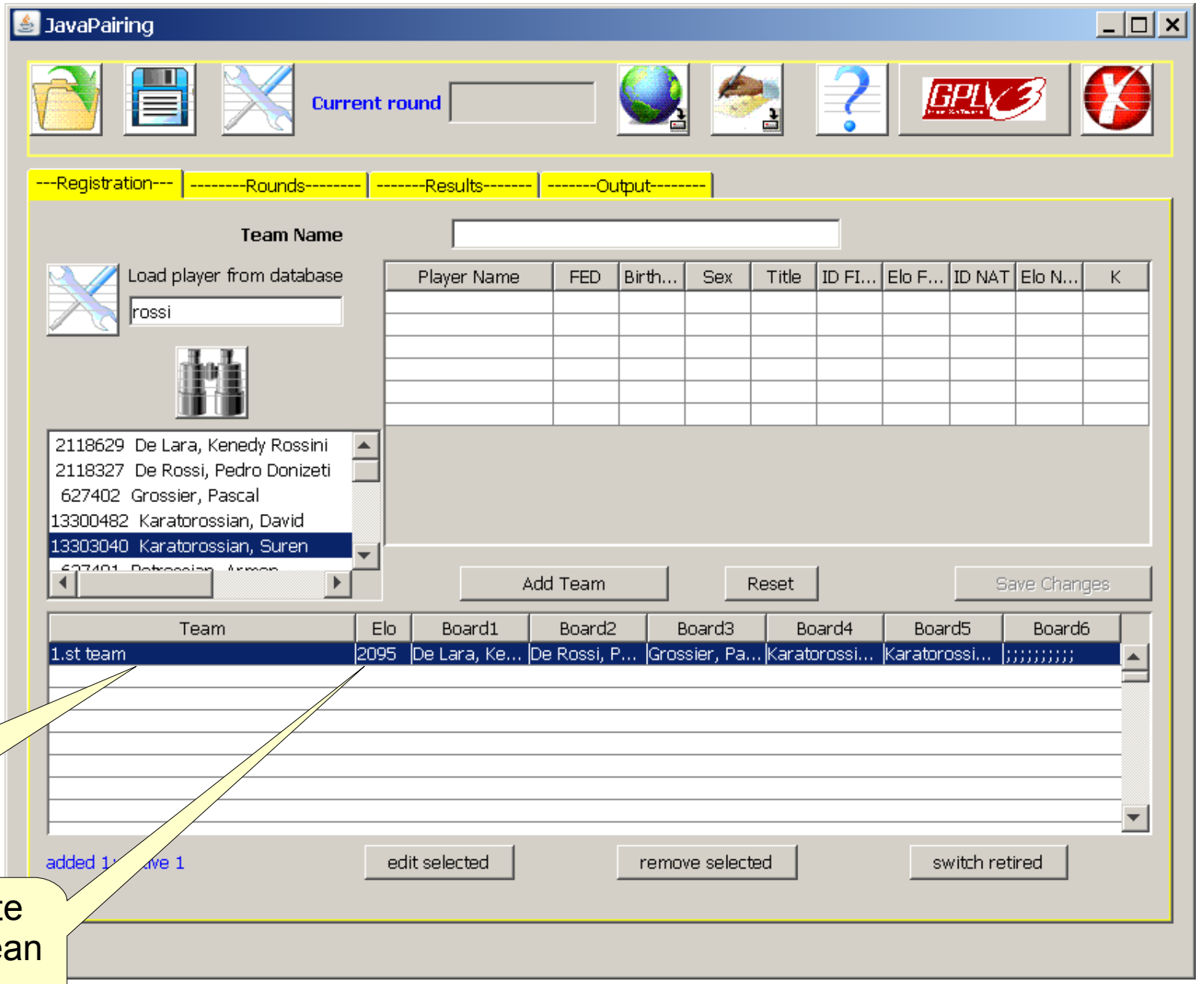
... type in the name of the Team ...

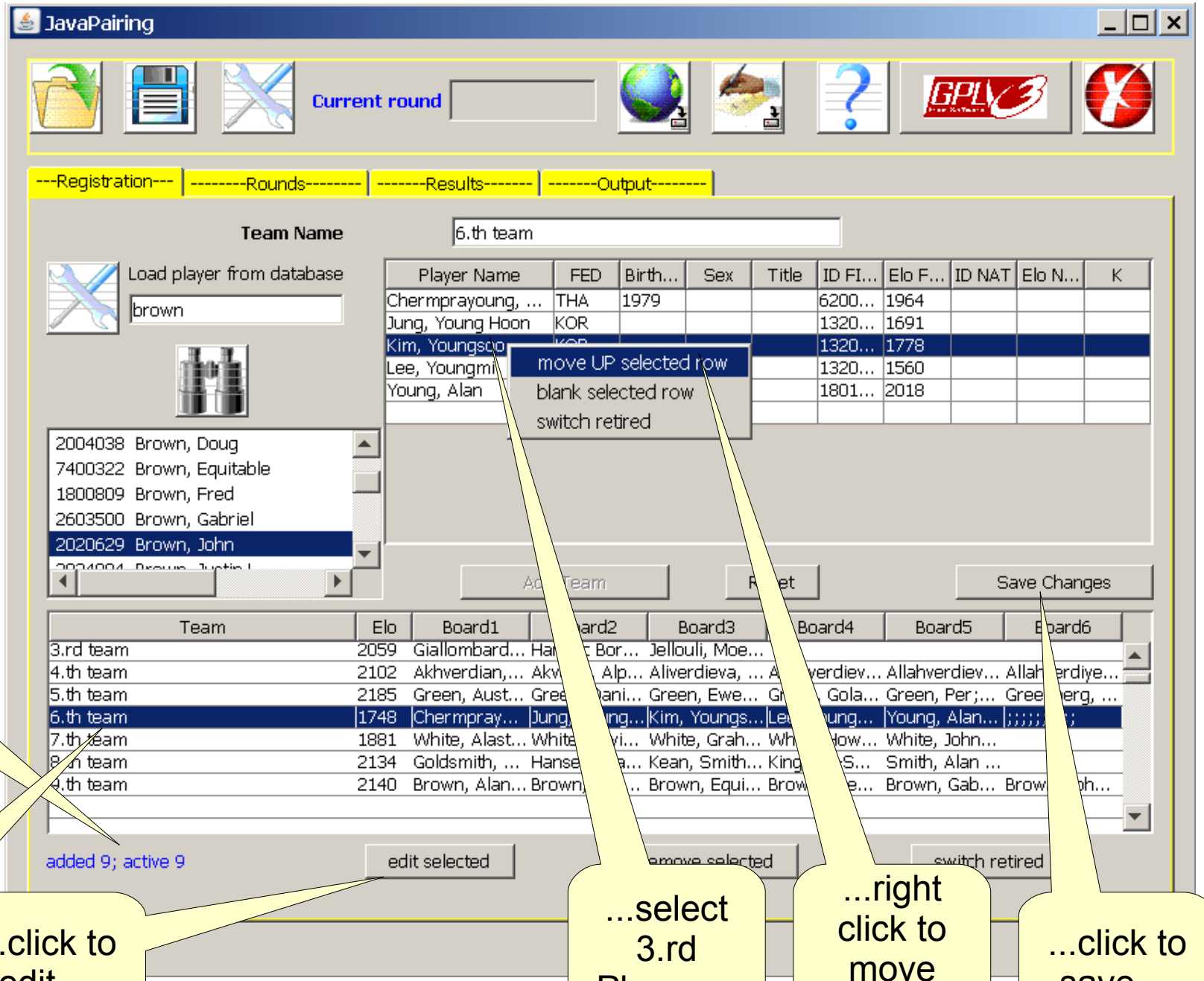
... ready to add Players ! ... type in part of the name to search for (5 chars at least) ...

... do search or press <Enter>...

... double click player or drag&drop into the grid ...

... when finished click to add Team ...


JavaPairing
Current round
GPL v3
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Team Name
Load player from database
rossi
Player Name
FED
Birth...
Sex
Title
ID FI...
Elo F...
ID NAT
Elo N...
K
2118629 De Lara, Kenedy Rossini
2118327 De Rossi, Pedro Donizeti
627402 Grossier, Pascal
13300482 Karatorossian, David
13303040 Karatorossian, Suren
Add Team
Reset
Save Changes
Team
Elo
Board1
Board2
Board3
Board4
Board5
Board6
1.st team
2095
De Lara, Ke...
De Rossi, P...
Grossier, Pa...
Karatorossi...
Karatorossi...
edit selected
remove selected
switch retired
... Team added! ...
... note the mean Elo ...



... 9 Teams added! ...

...select 6.th Team ...

...click to edit ...

...select 3.rd Player ...

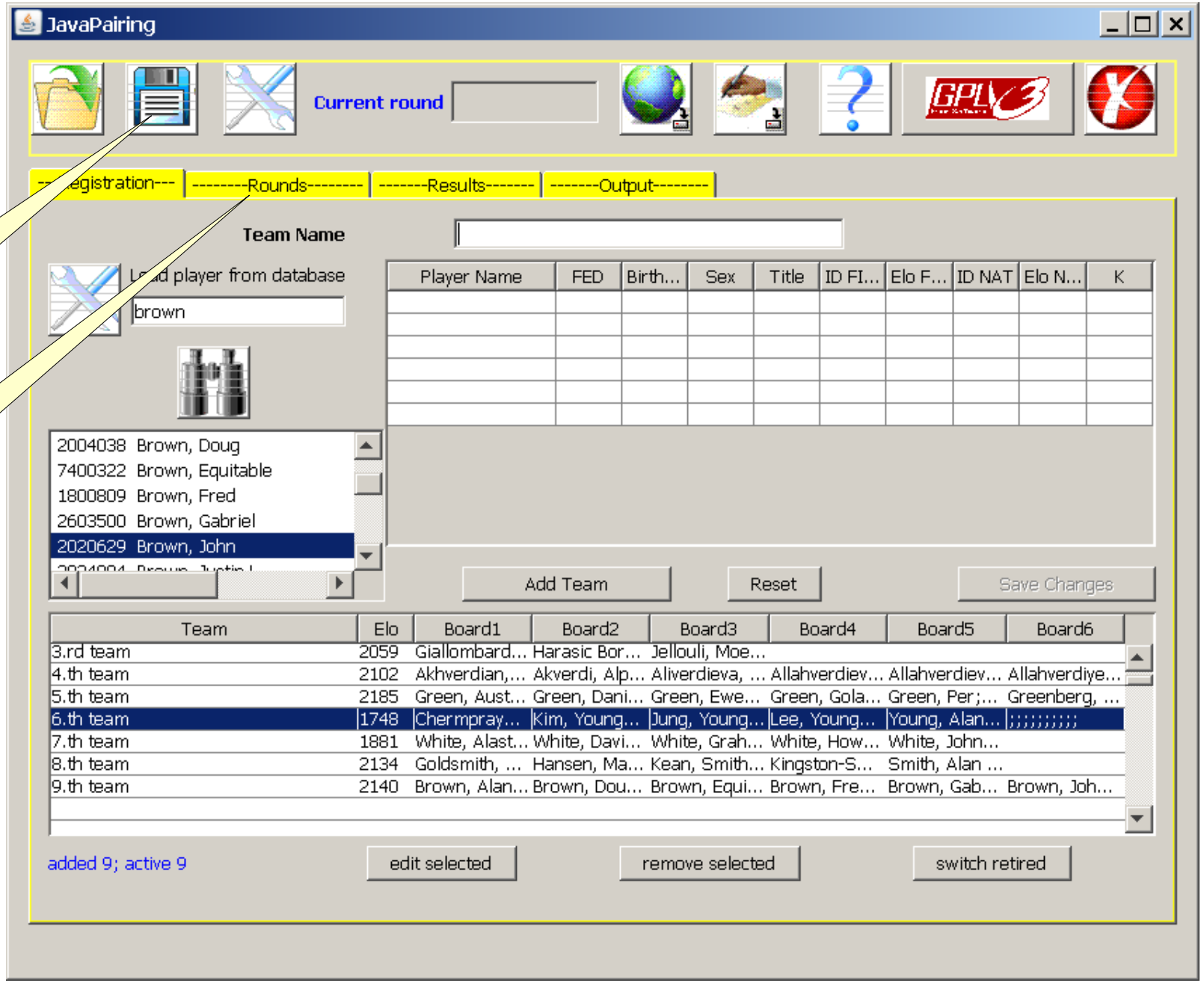
...right click to move up ...

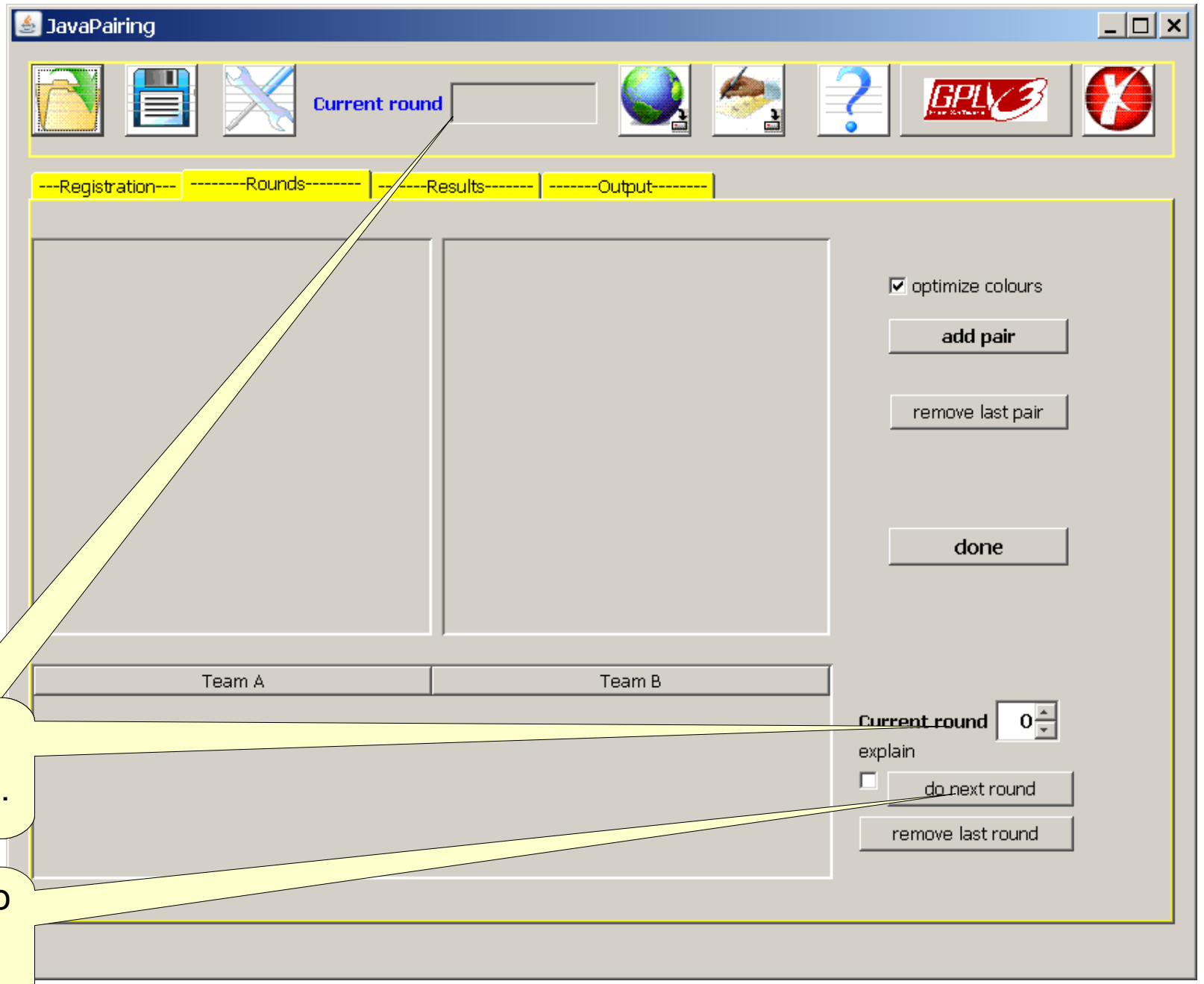
...click to save ...

... and so on...

... save to disc file...

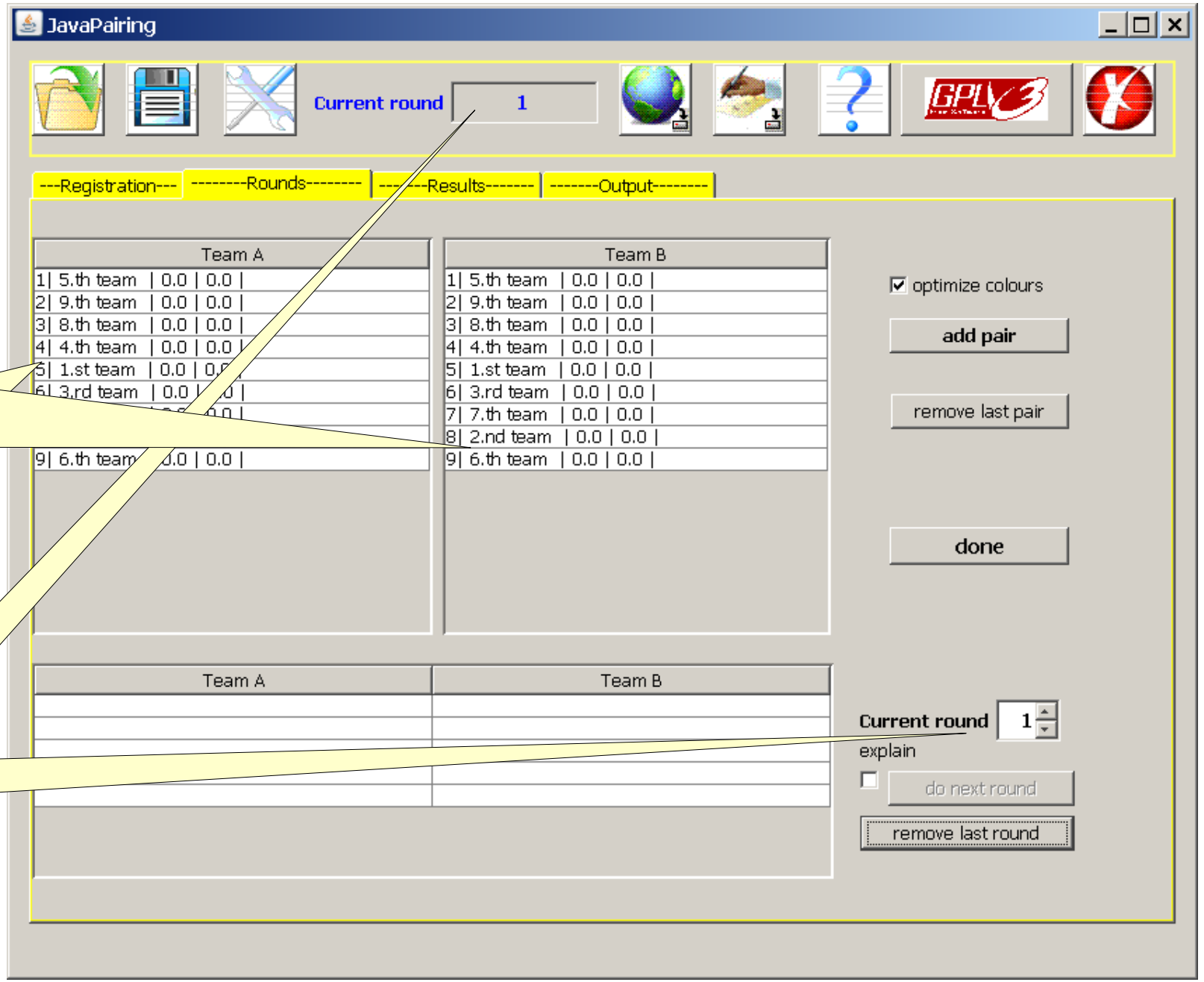
... go to rounds...

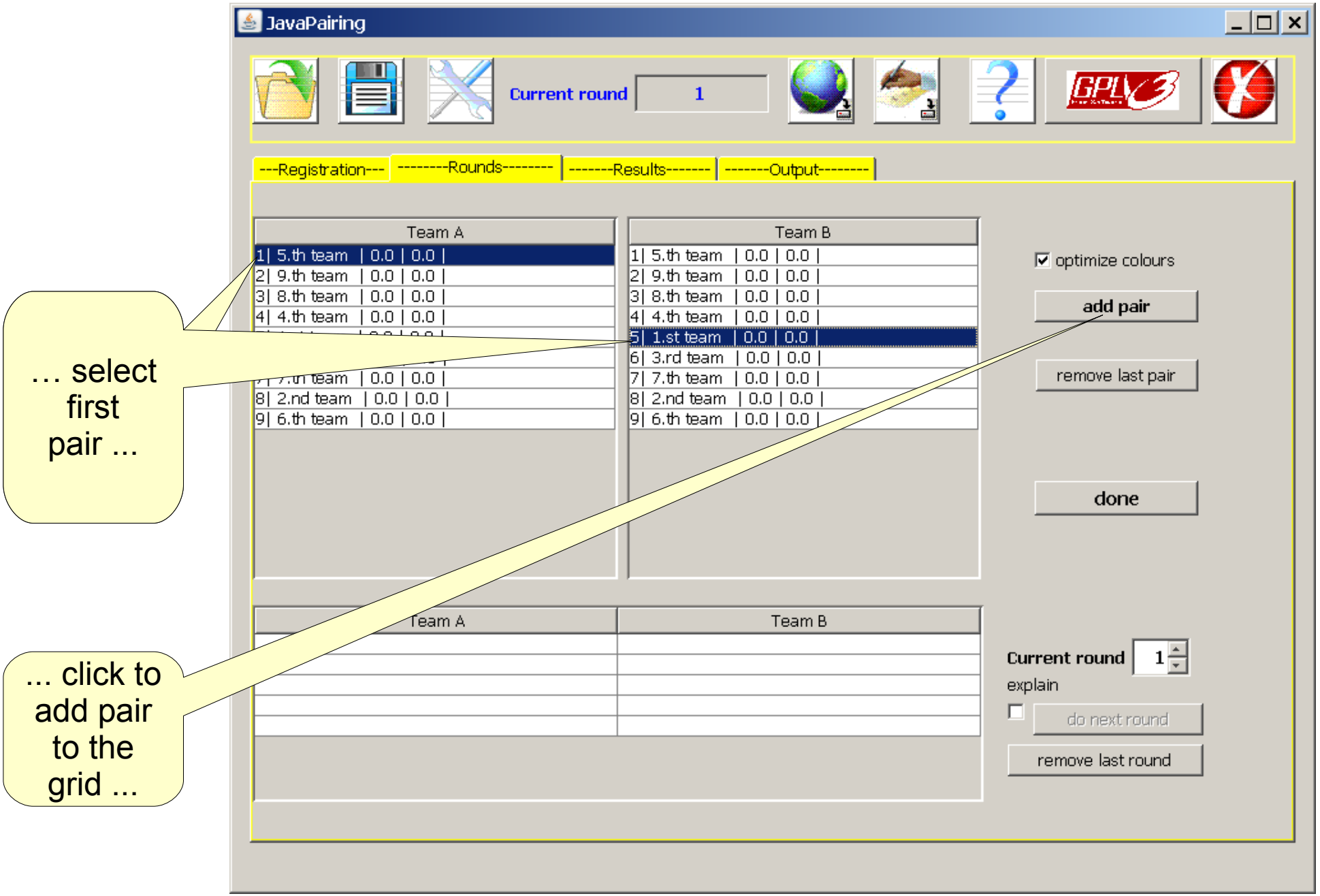


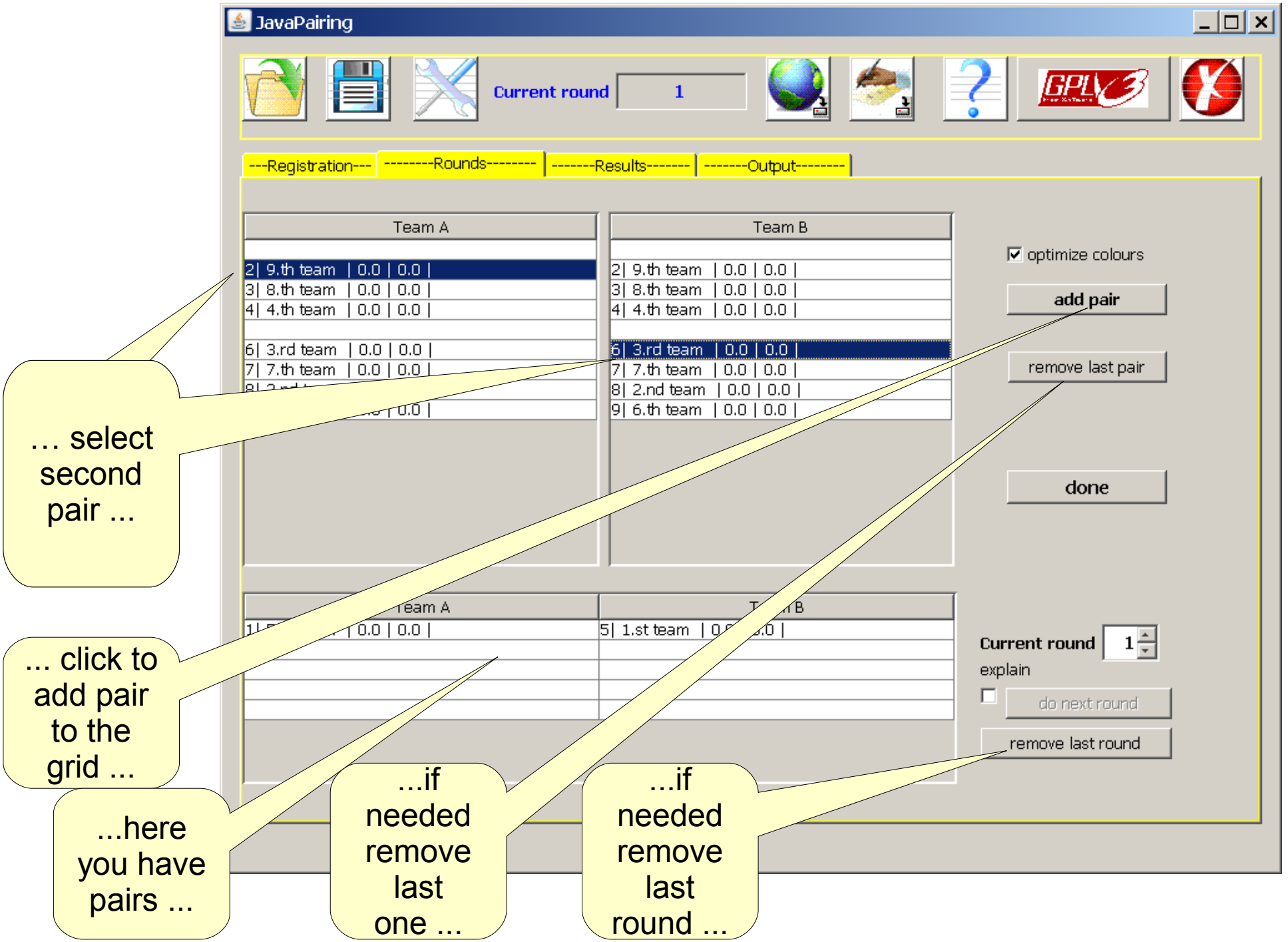


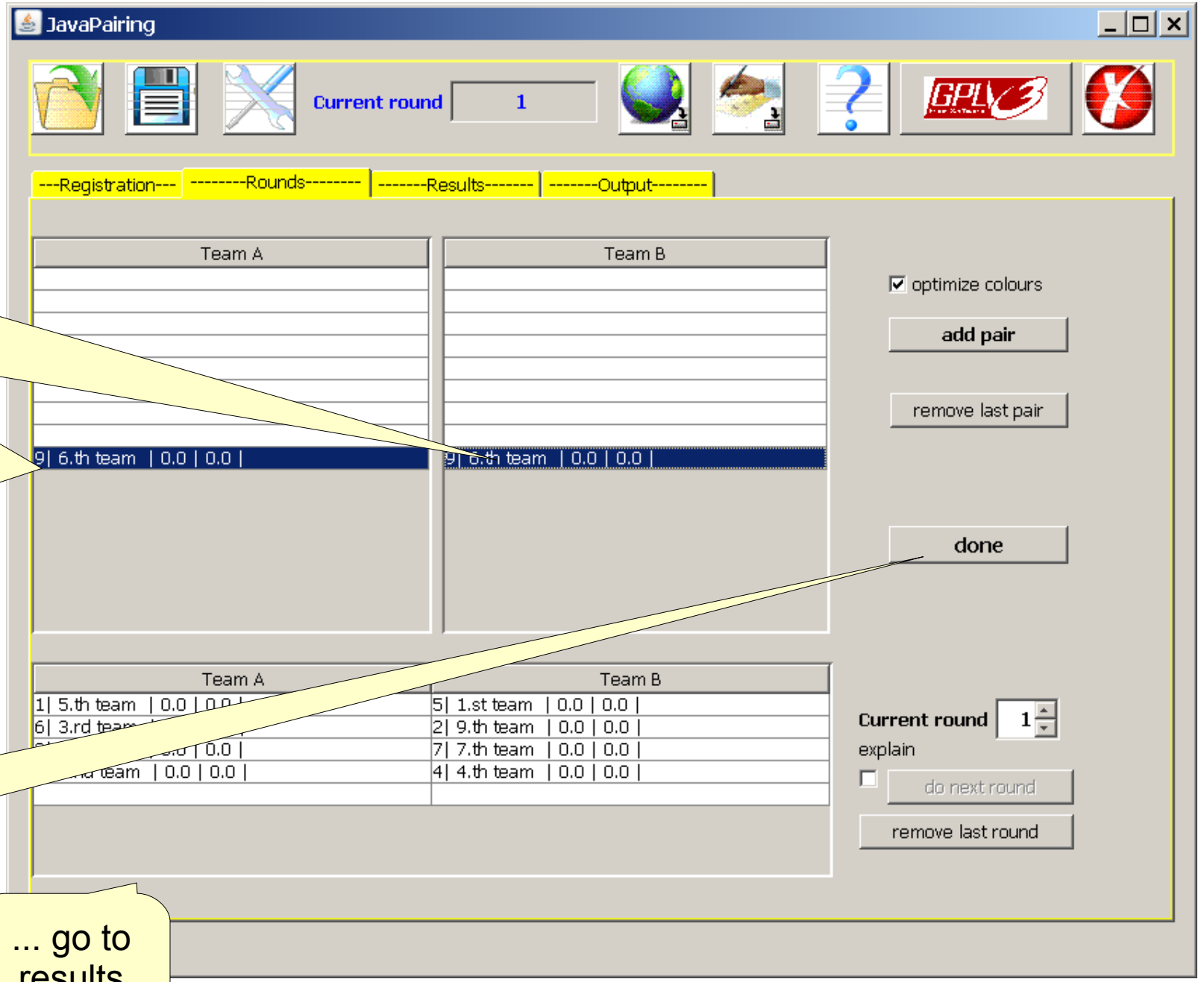
... round counter...

... click to do next round ...


JavaPairing
Current round
1
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Team A
Team B
1| 5.th team | 0.0 | 0.0 |
2| 9.th team | 0.0 | 0.0 |
3| 8.th team | 0.0 | 0.0 |
4| 4.th team | 0.0 | 0.0 |
5| 1.st team | 0.0 | 0.0 |
6| 3.rd team | 0.0 | 0.0 |
7| 7.th team | 0.0 | 0.0 |
8| 2.nd team | 0.0 | 0.0 |
9| 6.th team | 0.0 | 0.0 |
optimize colours
add pair
remove last pair
done
Team A
Team B
Current round
1
explain
do next round
remove last round
... Teams ordered decr. Elo ...
... round counter...


JavaPairing
Current round
1
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Team A
Team B
1| 5.th team | 0.0 | 0.0 |
2| 9.th team | 0.0 | 0.0 |
3| 8.th team | 0.0 | 0.0 |
4| 4.th team | 0.0 | 0.0 |
5| 1.st team | 0.0 | 0.0 |
6| 3.rd team | 0.0 | 0.0 |
7| 7.th team | 0.0 | 0.0 |
8| 2.nd team | 0.0 | 0.0 |
9| 6.th team | 0.0 | 0.0 |
optimize colours
add pair
remove last pair
done
Team A
Team B
Current round
1
explain
do next round
remove last round
… select first pair ...
... click to add pair to the grid ...


JavaPairing
Current round
1
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Team A
Team B
2| 9.th team | 0.0 | 0.0 |
3| 8.th team | 0.0 | 0.0 |
4| 4.th team | 0.0 | 0.0 |
6| 3.rd team | 0.0 | 0.0 |
7| 7.th team | 0.0 | 0.0 |
8| 2.nd team | 0.0 | 0.0 |
9| 6.th team | 0.0 | 0.0 |
5| 1.st team
optimize colours
add pair
remove last pair
done
Current round
1
explain
do next round
remove last round
… select second pair ...
... click to add pair to the grid ...
...here you have pairs ...
...if needed remove last one ...
...if needed remove last round ...

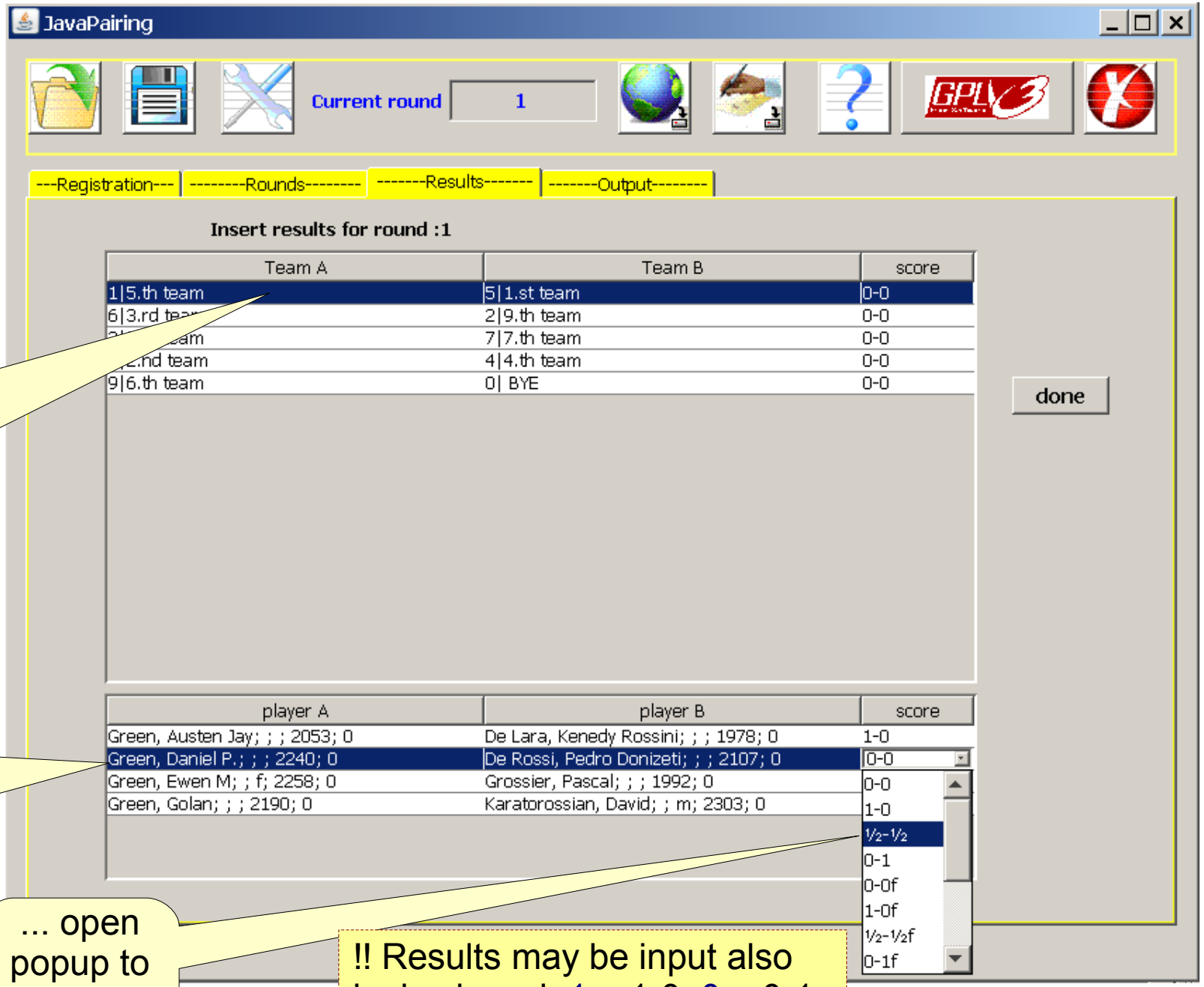


… select last pair because odd ...

… with itself means BYE ...

... click to finish and ...

... go to results page ..

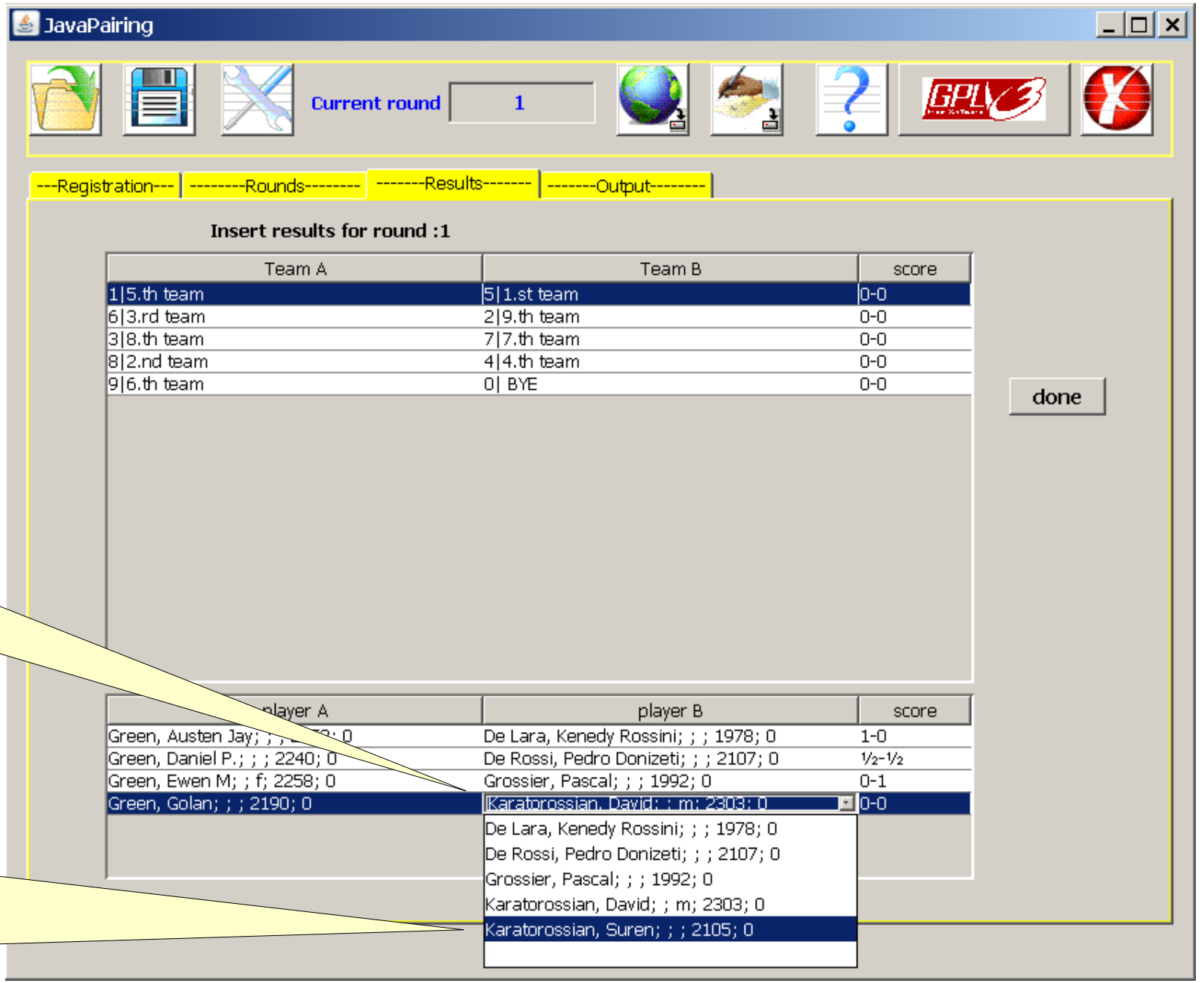


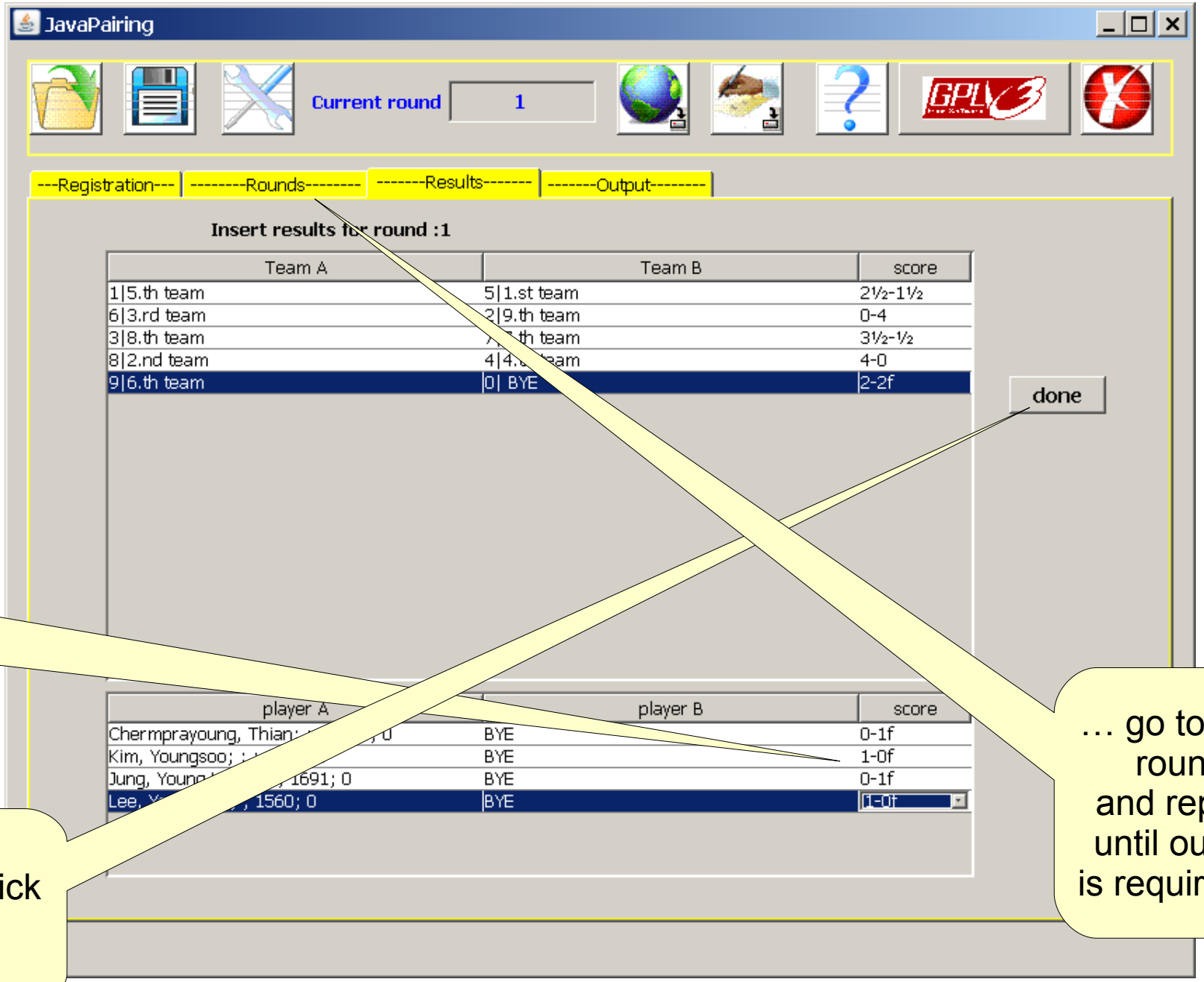
… click a pair and ...

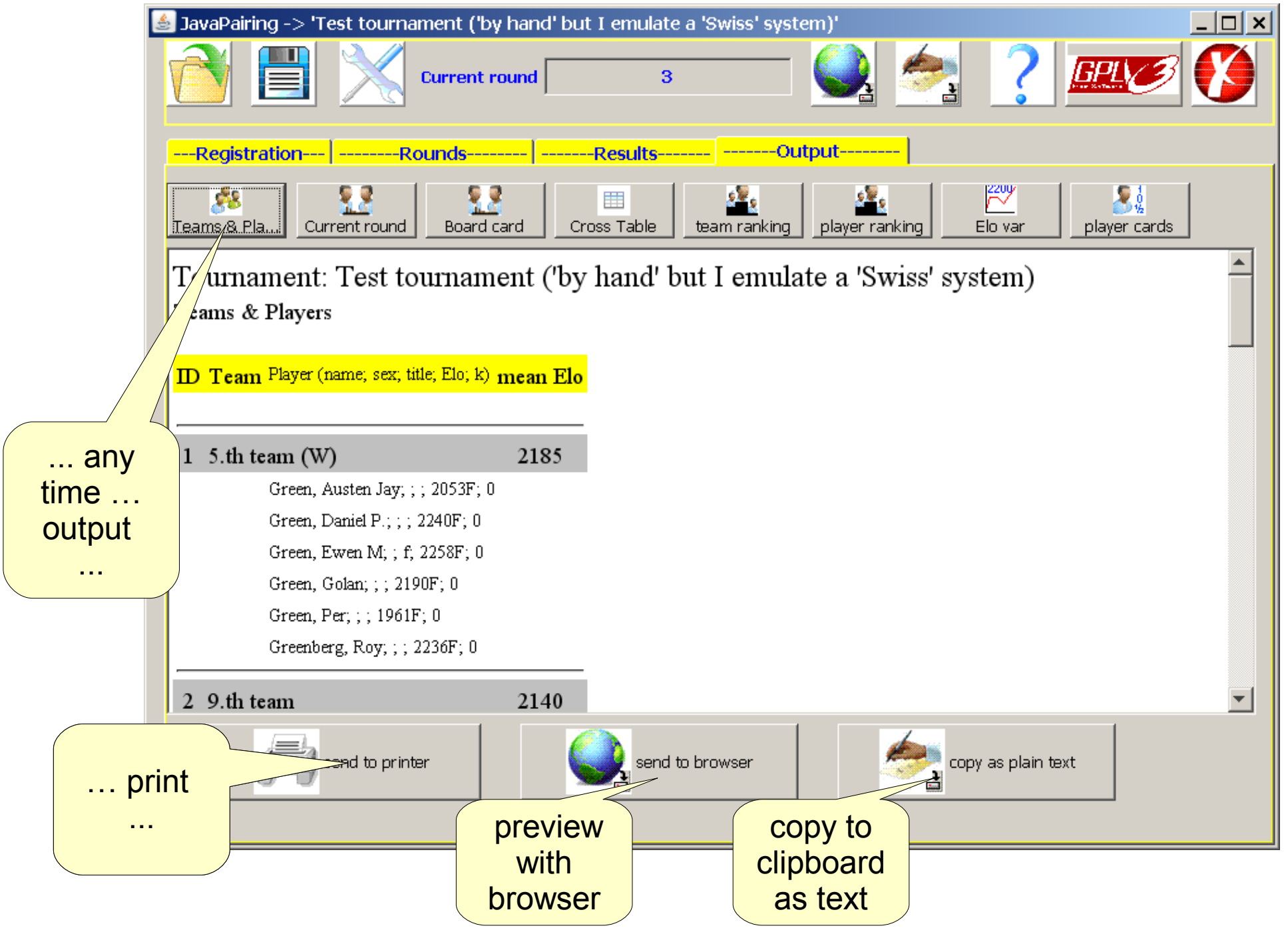
... click board ...

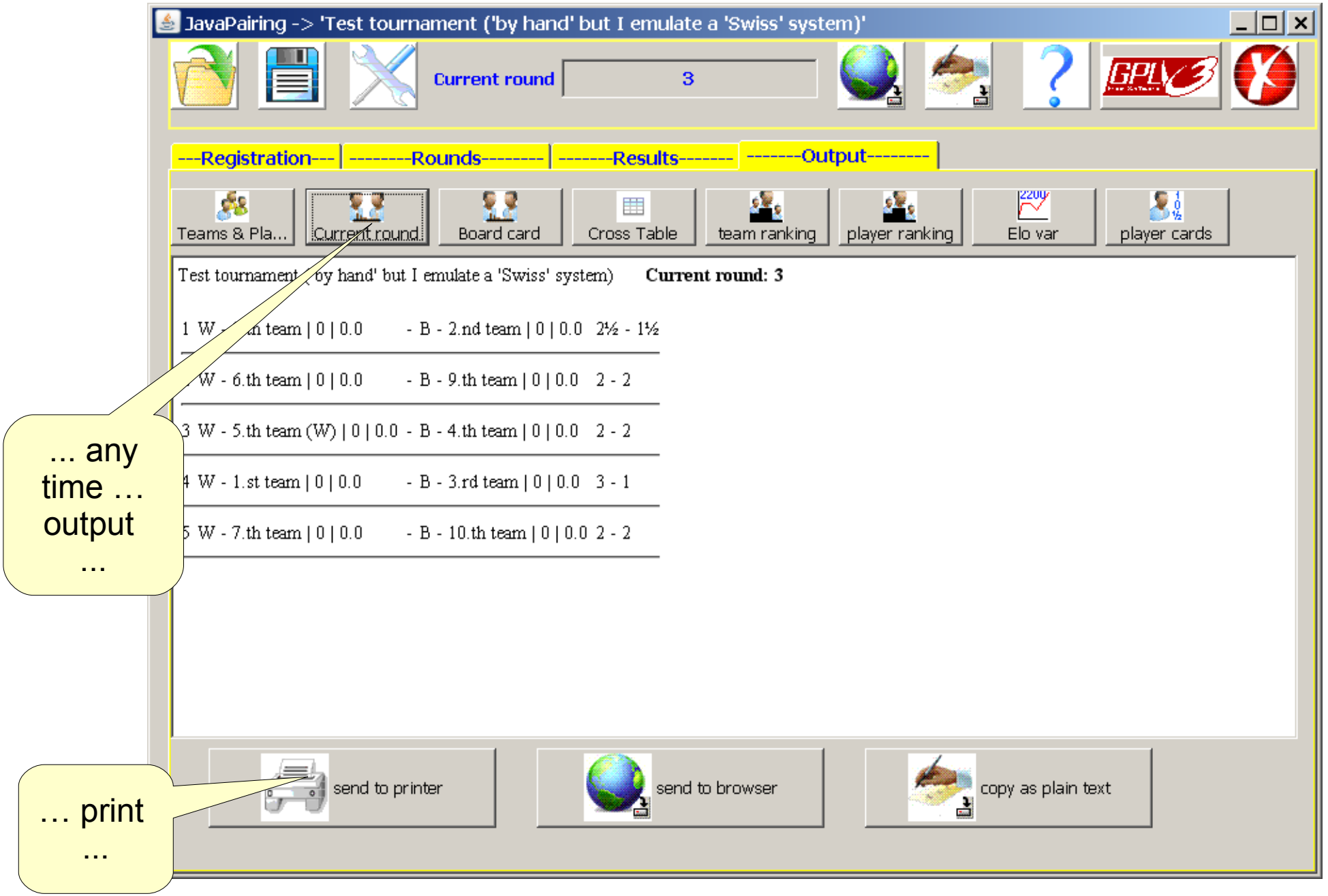
... open popup to select result ...

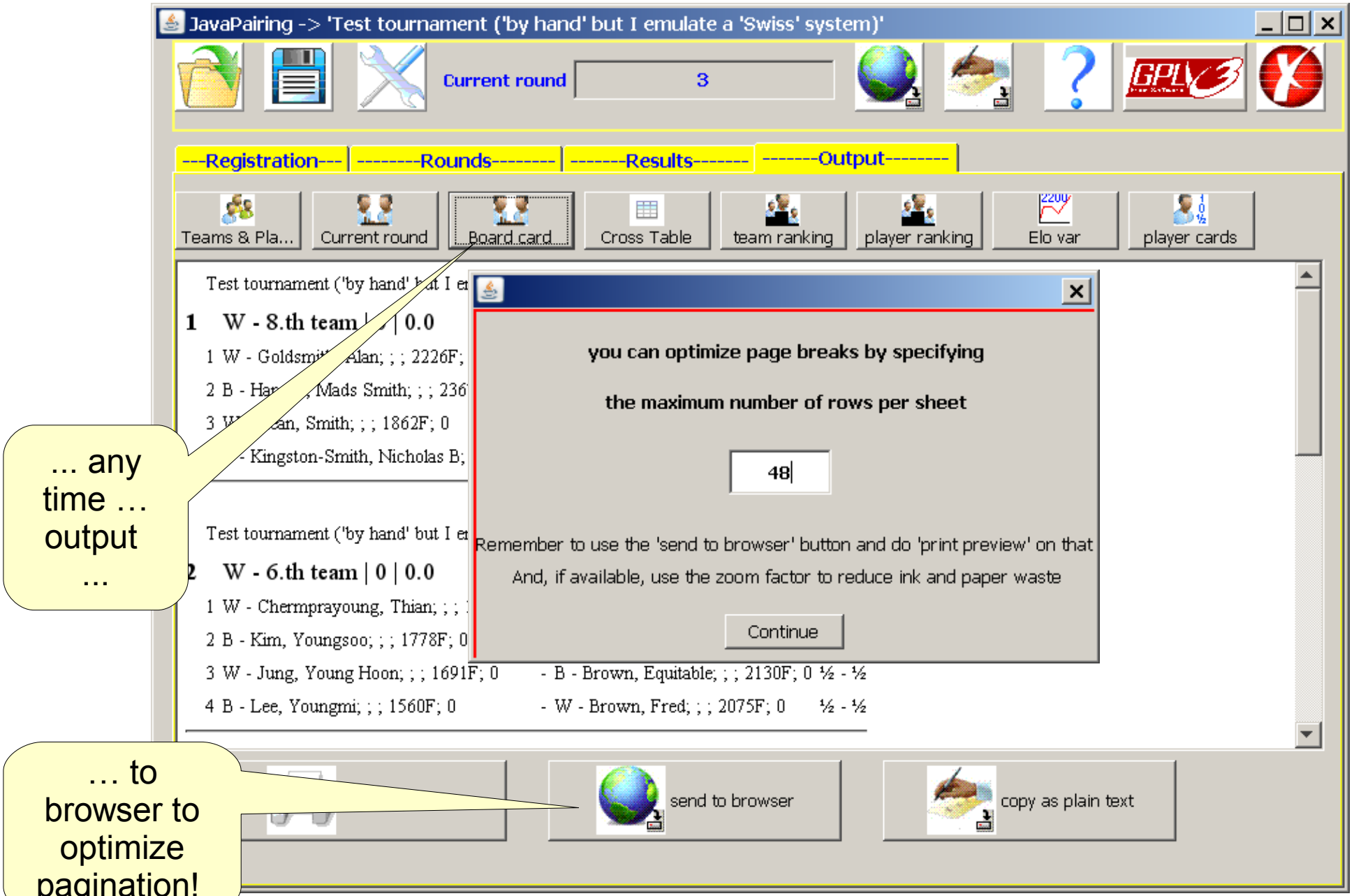
!! Results may be input also by keyboard: 1 = 1-0, 0 = 0-1, x or 5 = draw !!


JavaPairing
Current round
1
GPL v3
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Insert results for round :1
Team A
Team B
score
1|5.th team
5|1.st team
0-0
6|3.rd team
2|9.th team
0-0
3|8.th team
7|7.th team
0-0
8|2.nd team
4|4.th team
0-0
9|6.th team
0| BYE
0-0
done
player A
player B
score
Green, Austen Jay; ; ; 0
De Lara, Kenedy Rossini; ; ; 1978; 0
1-0
Green, Daniel P.; ; ; 2240; 0
De Rossi, Pedro Donizeti; ; ; 2107; 0
½-½
Green, Ewen M; ; f; 2258; 0
Grossier, Pascal; ; ; 1992; 0
0-1
Green, Golan; ; ; 2190; 0
Karatorossian, David; ; m; 2303; 0
0-0
De Lara, Kenedy Rossini; ; ; 1978; 0
De Rossi, Pedro Donizeti; ; ; 2107; 0
Grossier, Pascal; ; ; 1992; 0
Karatorossian, David; ; m; 2303; 0
Karatorossian, Suren; ; ; 2105; 0
... open popup to select a reserve ...
... you are free to change board order too ...


JavaPairing
Current round
1
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Insert results for round :1
Team A
Team B
score
1|5.th team
5|1.st team
2½-1½
6|3.rd team
2|9.th team
0-4
3|8.th team
th team
3½-½
8|2.nd team
4|4.
eam
4-0
9|6.th team
0| BYE
2-2f
done
player A
player B
score
Chermprayoung, Thian
0
BYE
0-1f
Kim, Youngsoo;
BYE
1-0f
Jung, Young
1691; 0
BYE
0-1f
Lee,
1560; 0
BYE
1-0f
... decide how to score the BYE ...
...when finished click done ...
… go to next round and repeat until output is required ...

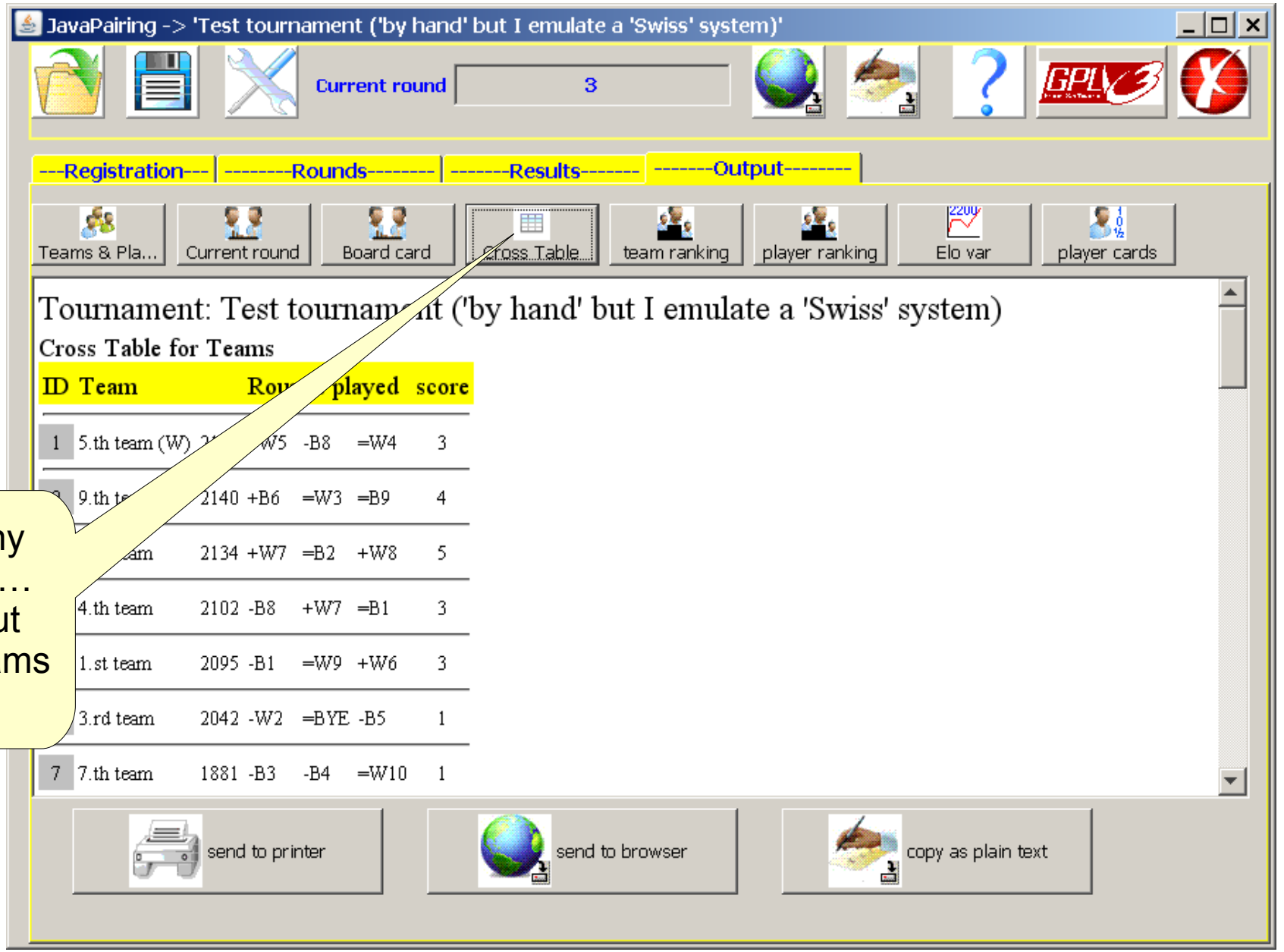

JavaPairing -> 'Test tournament ('by hand' but I emulate a 'Swiss' system)'
Current round
3
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Teams & Pla...
Current round
Board card
Cross Table
team ranking
player ranking
Elo var
player cards
Tournament: Test tournament ('by hand' but I emulate a 'Swiss' system)
Teams & Players
ID Team Player (name; sex; title; Elo; k) mean Elo
1 5.th team (W) 2185
Green, Austen Jay; ; ; 2053F; 0
Green, Daniel P.; ; ; 2240F; 0
Green, Ewen M; ; f; 2258F; 0
Green, Golan; ; ; 2190F; 0
Green, Per; ; ; 1961F; 0
Greenberg, Roy; ; ; 2236F; 0
2 9.th team 2140
send to printer
send to browser
copy as plain text
... any time … output ...
… print ...
preview with browser
copy to clipboard as text


JavaPairing -> 'Test tournament ('by hand' but I emulate a 'Swiss' system)'
Current round
3
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Teams & Pla...
Current round
Board card
Cross Table
team ranking
player ranking
Elo var
player cards
Test tournament ('by hand' but I emulate a 'Swiss' system)
Current round: 3
1 W - h team | 0 | 0.0 - B - 2.nd team | 0 | 0.0 2½ - 1½
W - 6.th team | 0 | 0.0 - B - 9.th team | 0 | 0.0 2 - 2
3 W - 5.th team (W) | 0 | 0.0 - B - 4.th team | 0 | 0.0 2 - 2
4 W - 1.st team | 0 | 0.0 - B - 3.rd team | 0 | 0.0 3 - 1
5 W - 7.th team | 0 | 0.0 - B - 10.th team | 0 | 0.0 2 - 2
send to printer
send to browser
copy as plain text
... any time … output ...
… print ...

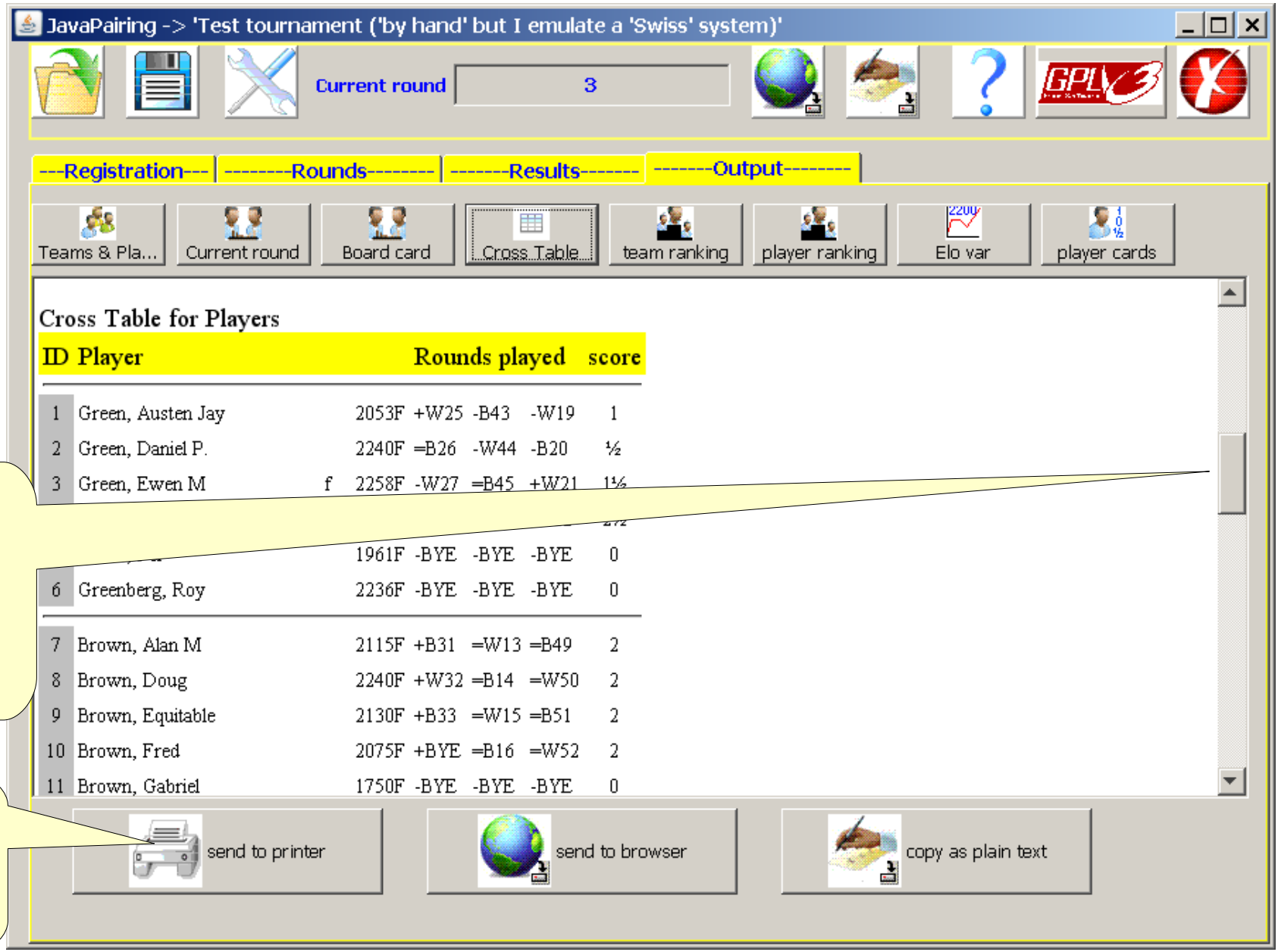


... any time … output ...

… to browser to optimize pagination!

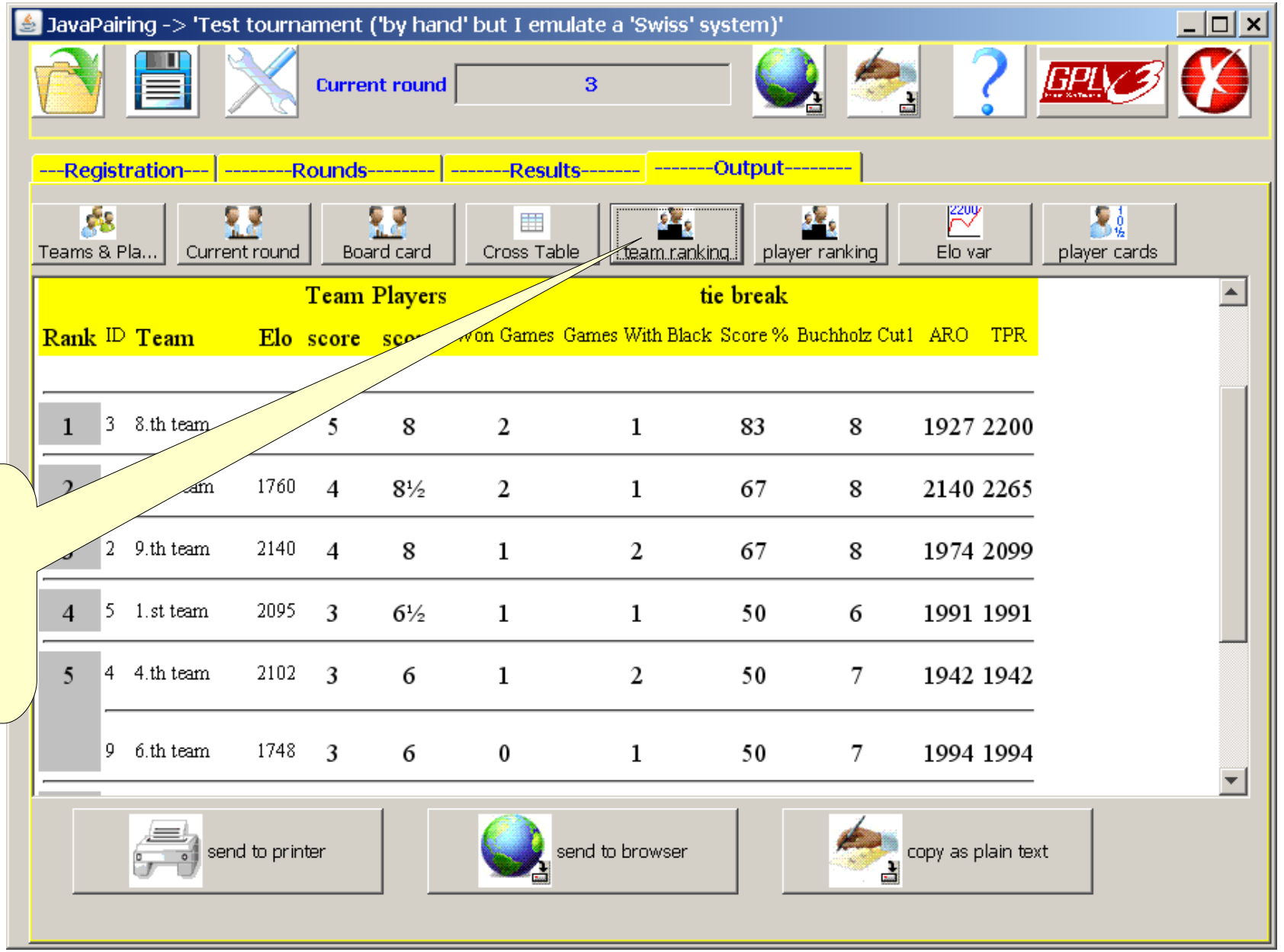


... any time … output for teams ...



… scroll for players …

… print …



... any time … output team ranking ...

... any time … output player ranking ...

JavaPairing -> 'Test tournament ('by hand' but I emulate a 'Swiss' system)'

Current round 3

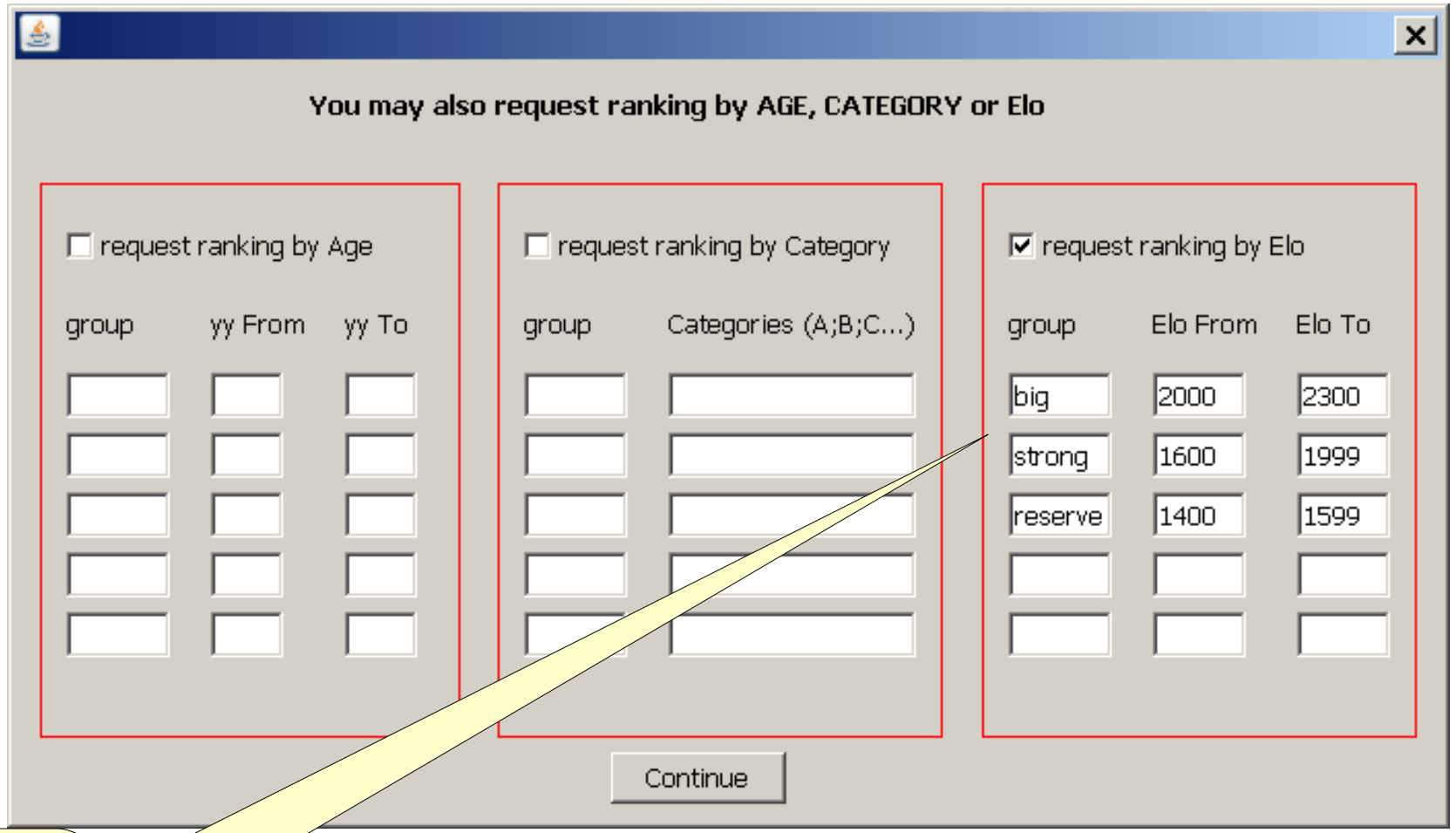
---Registration--- --------Rounds-------- -------Results------- -------Output--------

Teams & Pla... | Current round | Board card | Cross Table | team ranking | player ranking | Elo var | player cards

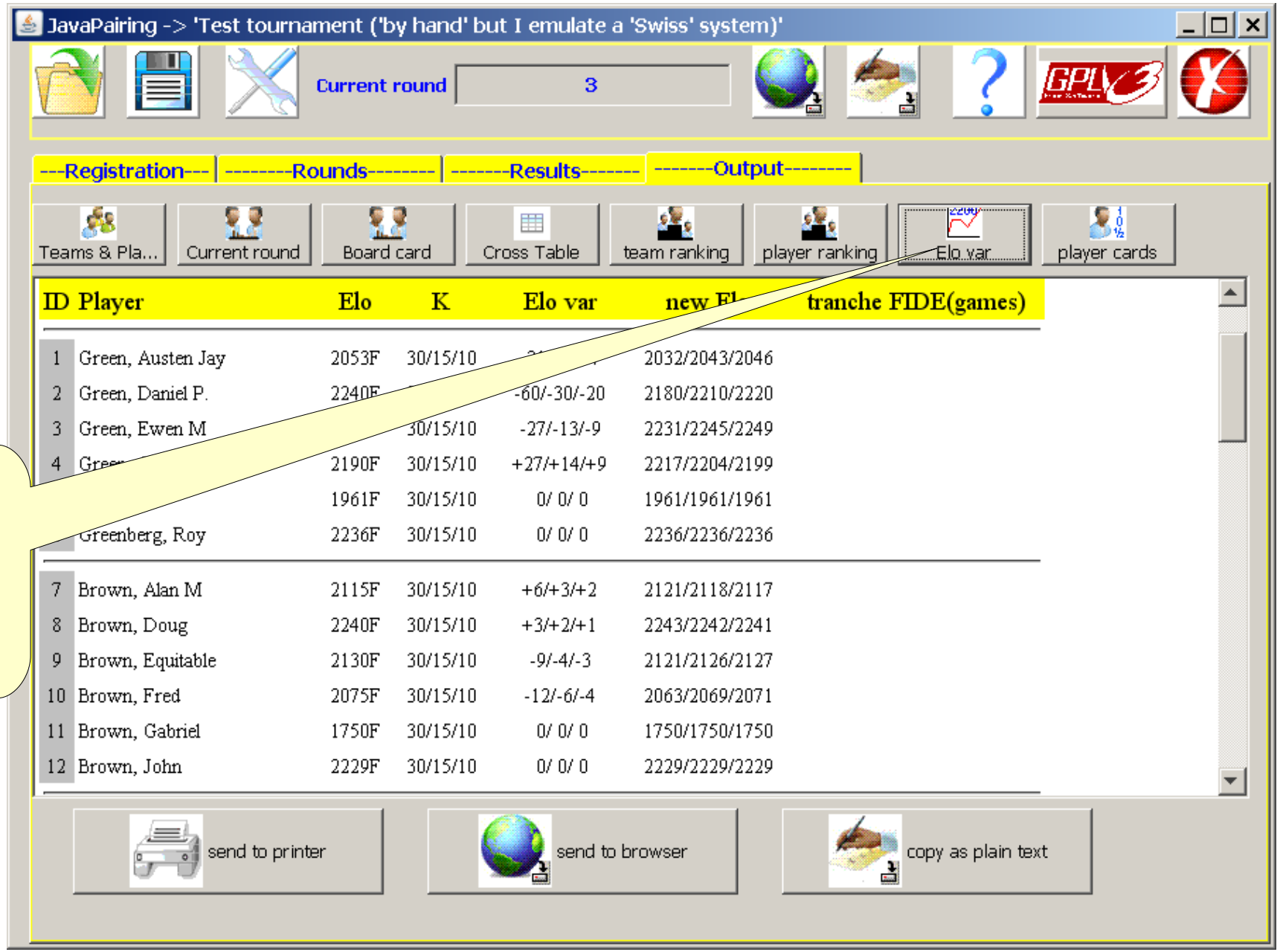
Board1:

Rank	ID	Player		score	Won Games	Games With Black	Score %	tie break Buchholz Cut1	ARO	TPR
1	43	A[illegible], Bianca	1963F	2½	2	1	83⅓	4	2105	2378
		Akhverdian, Gabriel	2038F	2	2	2	66⅔	3½	2022	2147
	7	Brown, Alan M	2115F	2	1	2	66⅔	3½	2040	2165
	13	Goldsmith, Alan	2226F	2	1	1	66⅔	4½	2043	2168
5	25	De Lara, Kenedy Rossini	1978F	1½	1	1	50	2½	1982	1982

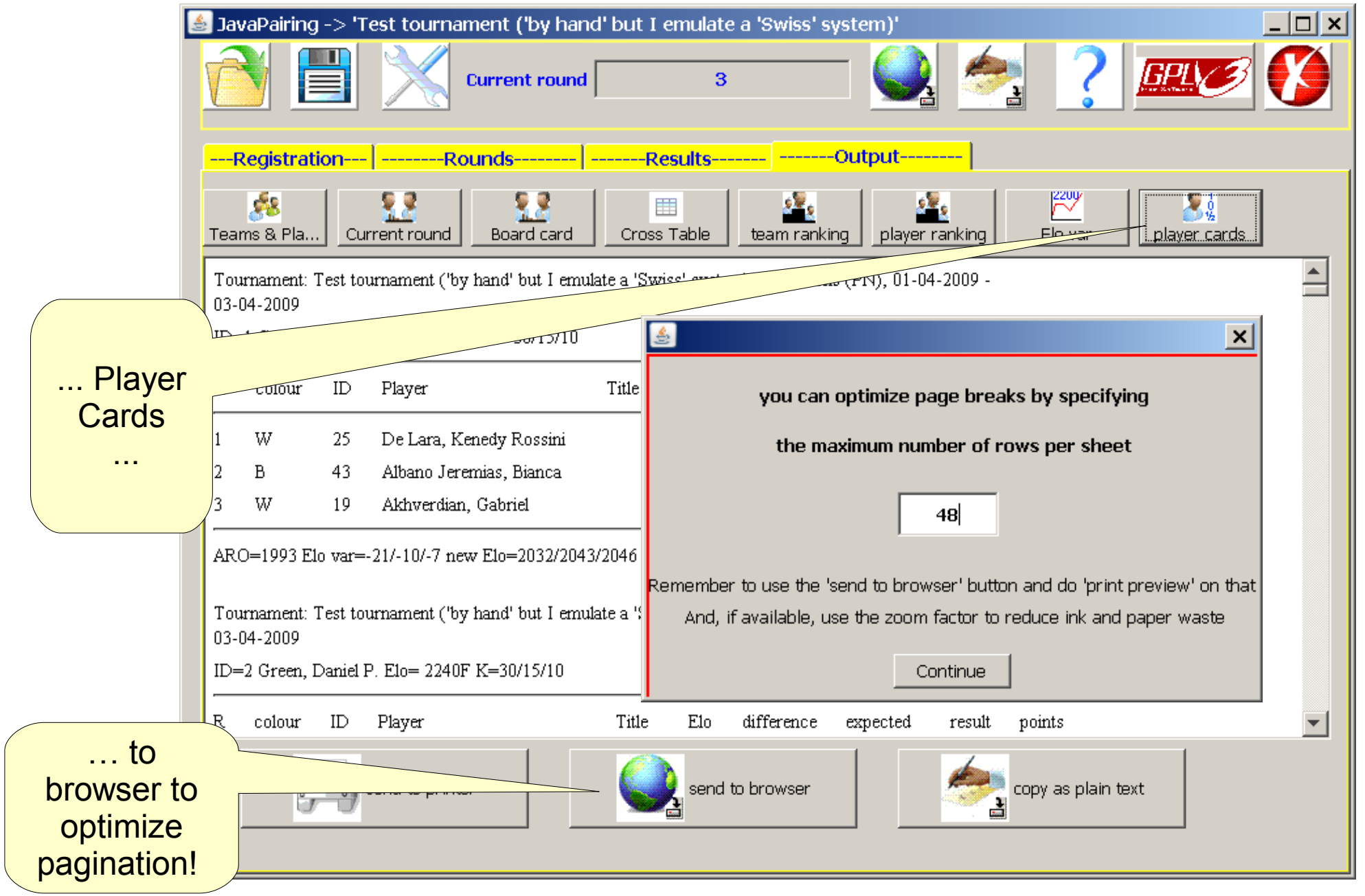
send to printer | send to browser | copy as plain text

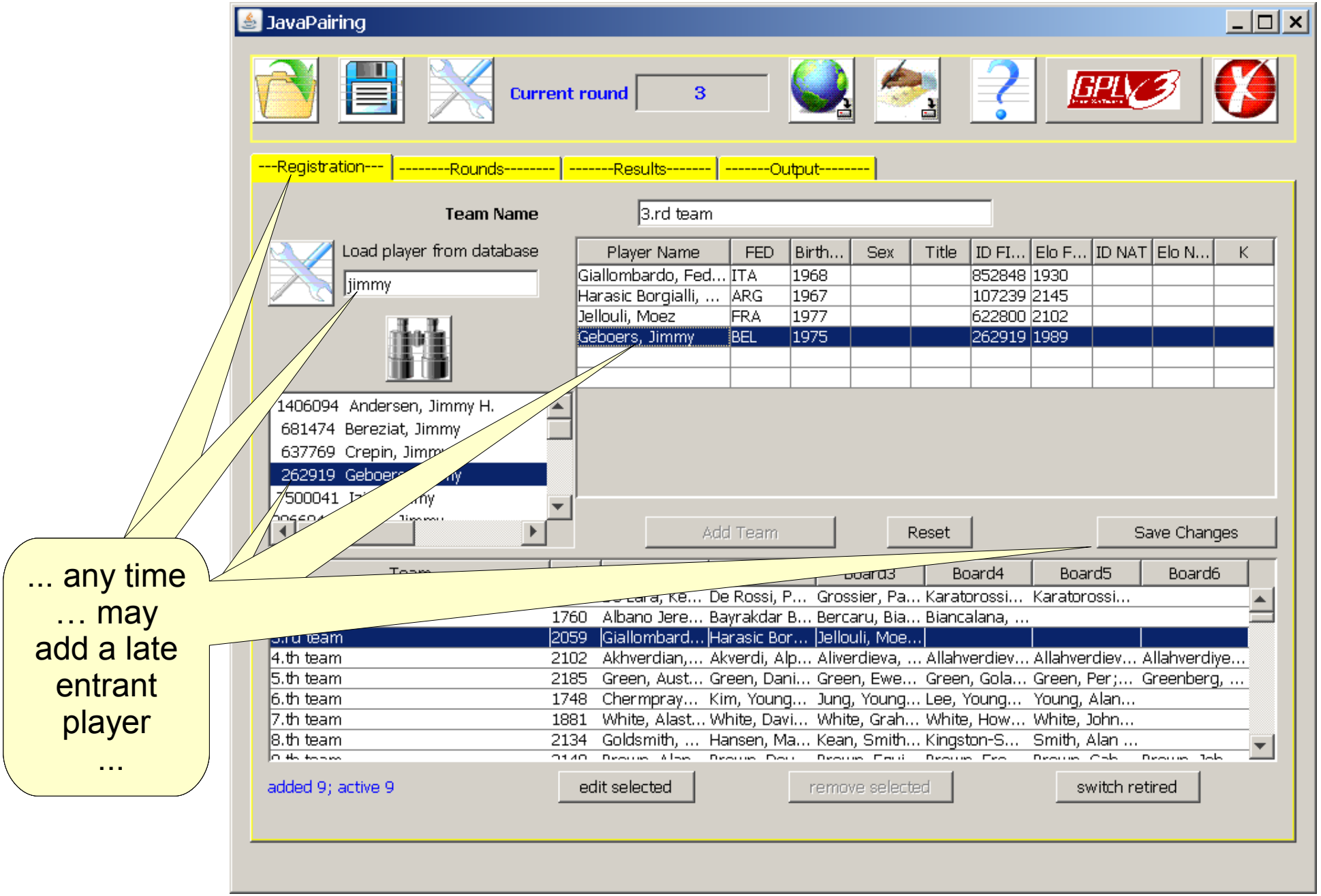


... partial ranking for teams & players is available...

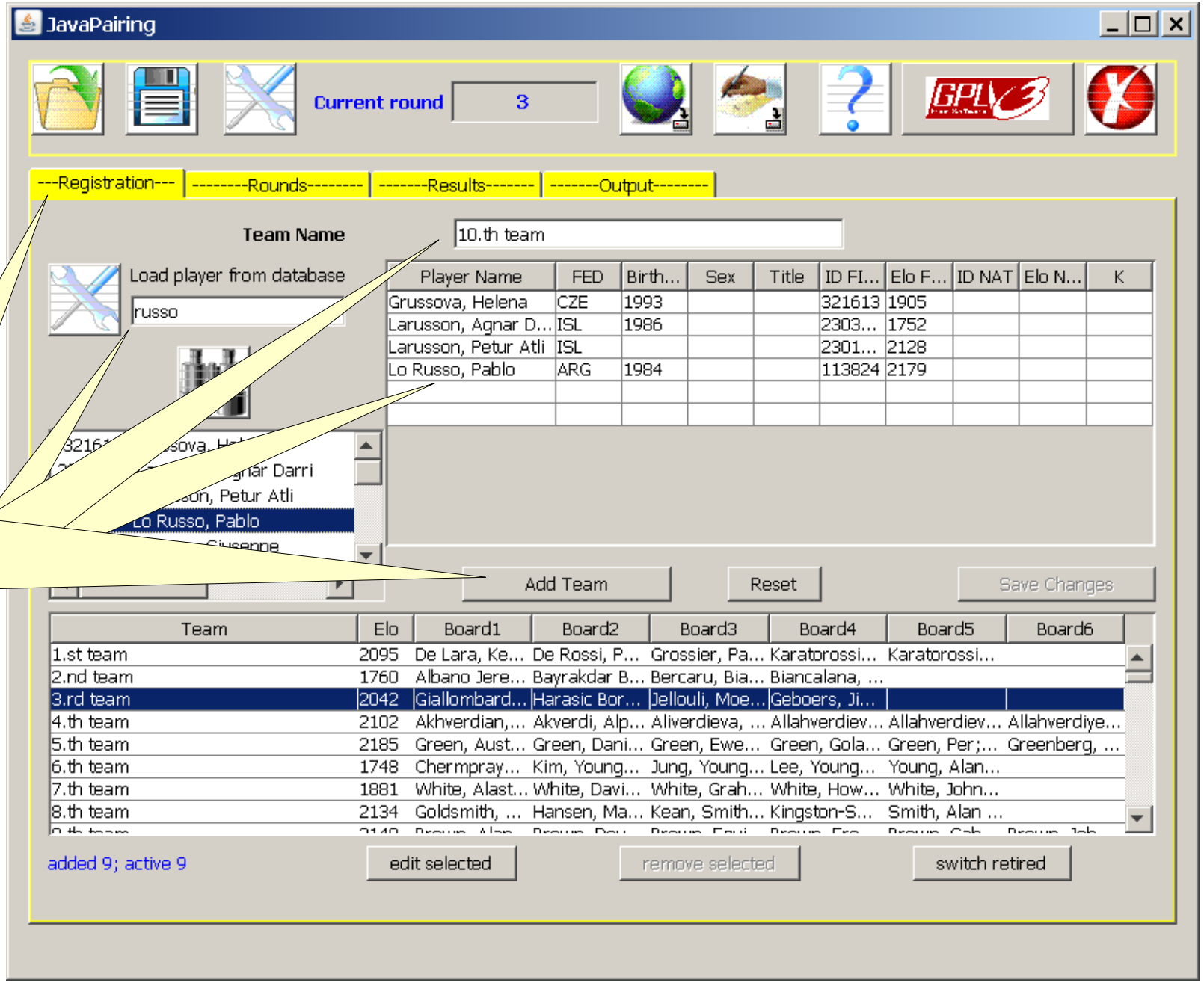


... Elo variation ...

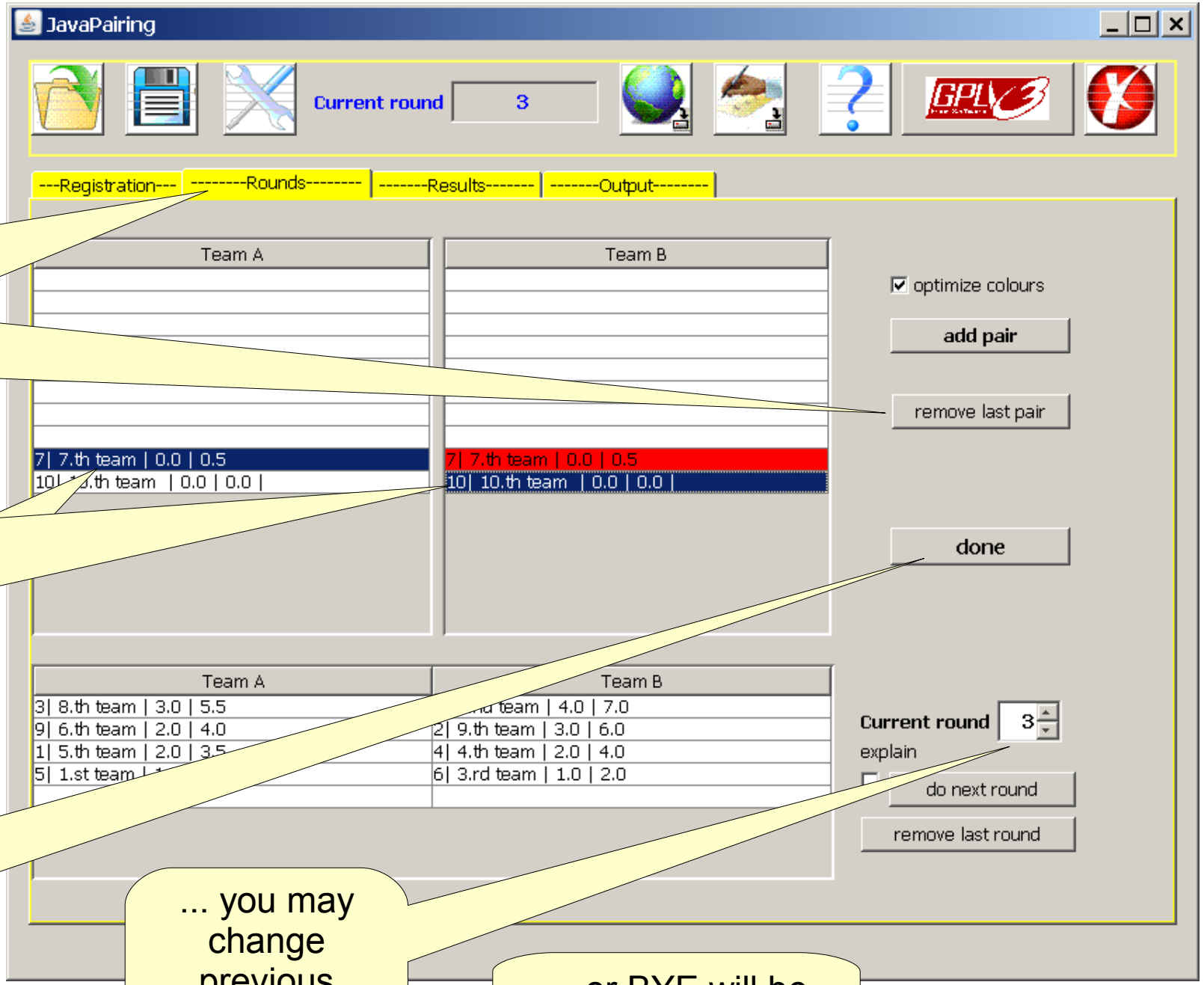

JavaPairing -> 'Test tournament ('by hand' but I emulate a 'Swiss' system)'
Current round
3
---Registration---
--------Rounds--------
-------Results-------
-------Output--------
Teams & Pla...
Current round
Board card
Cross Table
team ranking
player ranking
player cards
colour ID Player Title
1 W 25 De Lara, Kenedy Rossini
2 B 43 Albano Jeremias, Bianca
3 W 19 Akhverdian, Gabriel
ARO=1993 Elo var=-21/-10/-7 new Elo=2032/2043/2046
Tournament: Test tournament ('by hand' but I emulate a 'S
03-04-2009
ID=2 Green, Daniel P. Elo= 2240F K=30/15/10
R colour ID Player Title Elo difference expected result points
you can optimize page breaks by specifying
the maximum number of rows per sheet
48
Remember to use the 'send to browser' button and do 'print preview' on that
And, if available, use the zoom factor to reduce ink and paper waste
Continue
send to browser
copy as plain text
... Player Cards ...
… to browser to optimize pagination!



... any time
… may
add a late
entrant
player
...



... any time
… may
add a late
entrant
team
...



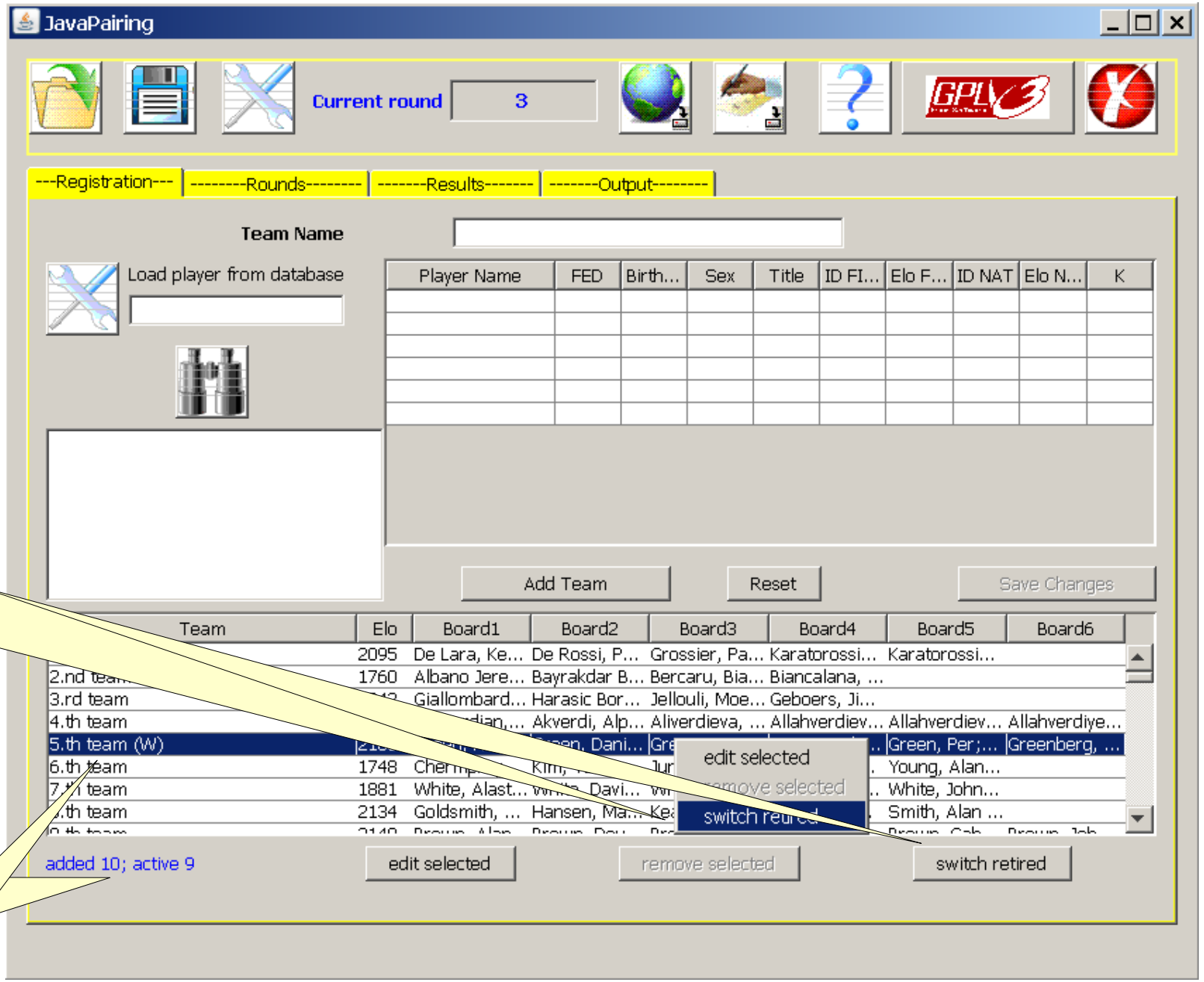
... remove last pair ...

... pair late entrant team ...

... add pair and click done ...

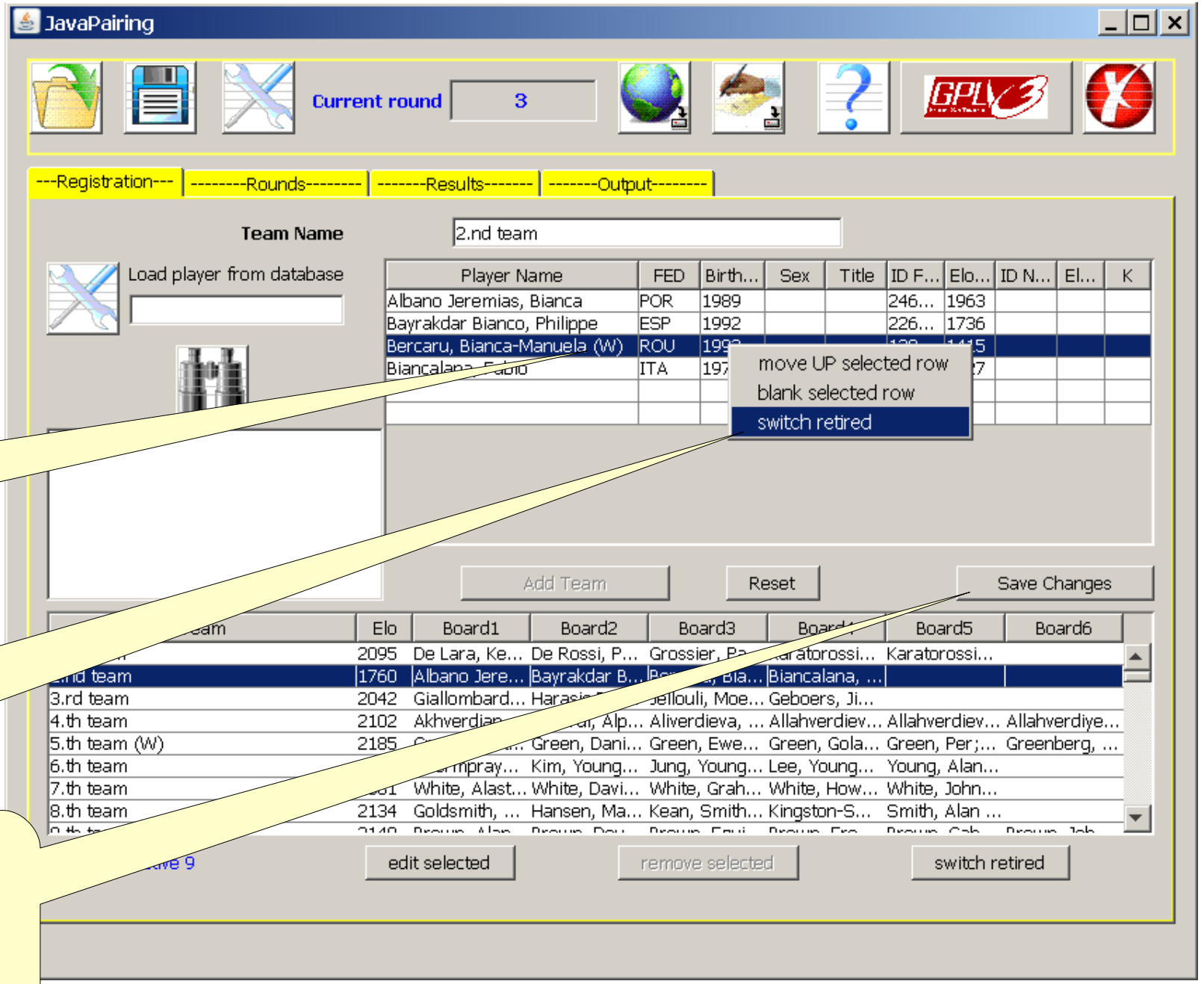
... you may change previous rounds as well ...

... or BYE will be assigned ...



... switch retired team ...

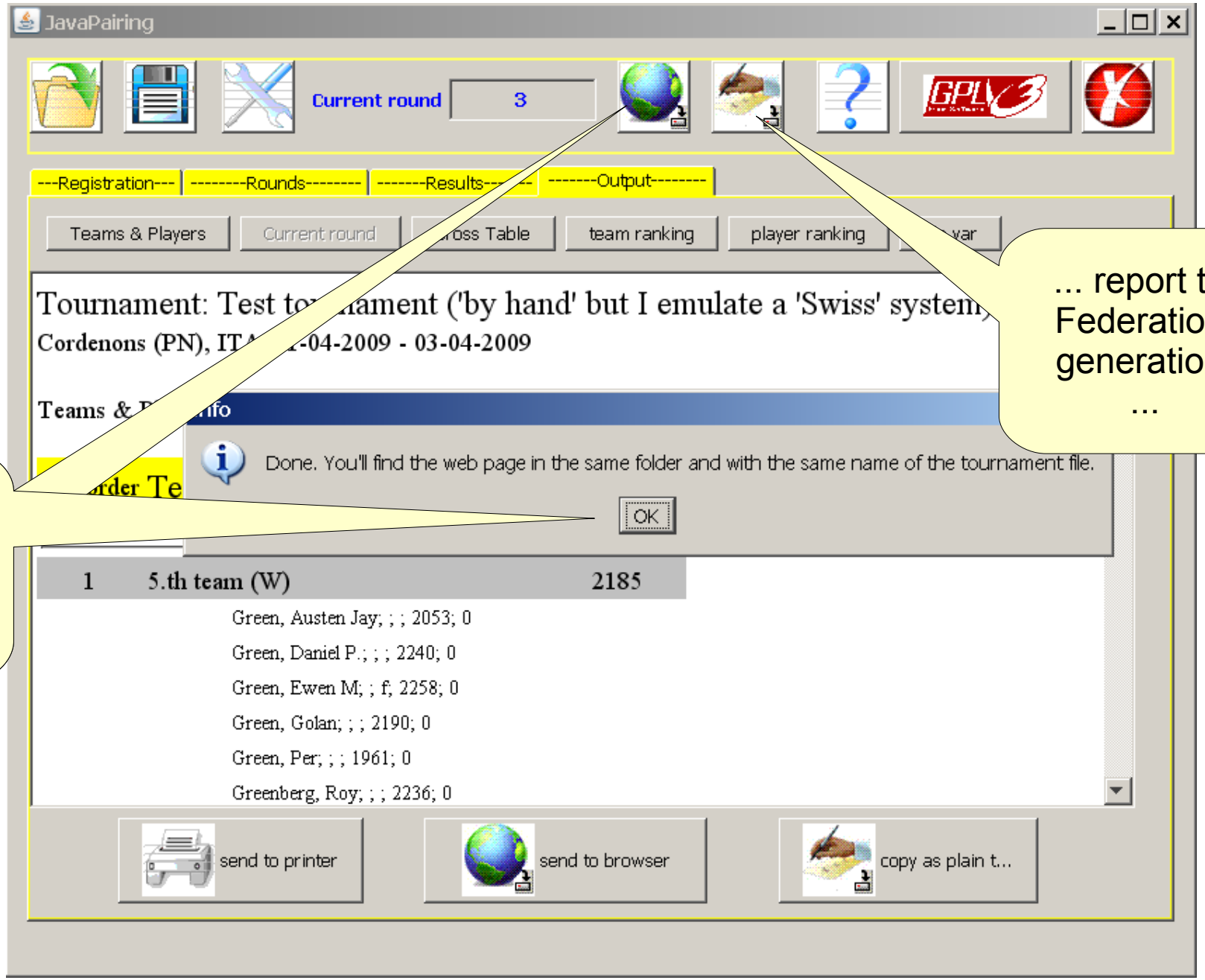
... marking retired team ...



... marking retired player ...

… switch retired player ...

… save changes ...



... WEB page generation ...

... report to Federation generation ...

Internationalization

When writing this manual, JavaPairing is already available in English, Italian, German and French languages. English is the native language for the source code. To test the correctness of translations or to evaluate the native interface, or simply out of curiosity, the program could be run by:
- default JavaPairing runs in the language selected in the control panel of the operating system, if already implemented, or otherwise in English. So in Italy it starts in Italian, in Germany in German, ecc ...
- to have another language, you may change settings in the control panel
- one of the alternatives is to run JavaPairing with two parameters, xx and YY, where xx is the language code and YY is the country code (it IT means Italian of Italy, de DE means German of Germany, etc...)
a) by creating a link on the desktop and adding the parameters at the source
b) by the command shell ("java -jar JavaPairing.jar xx YY"; be careful of case sensibility on some operating systems ...).
You may have all translations running at the same time! Translation happens by substituting strings on the fly at run time, taken from a plain text file named JavaPairing_xx.properties, which is packed into JavaPairing.jar. To make a new translation you may start from the English version, translate the right part of the rows and name the file with your xx. In the folder 'i18n' you will find the already translated versions and a list of all possible language and country codes.
Translators awarded! See the following page too.

n.b. When translating the strings of the user interface be careful to be short enough to not cause an unpleasant resize of the main window

Labels, menus, messages can be easily translated to other languages working on a text file

JavaPairing.properties

```
Reset                                = Reset
result                               = result
Results                              = -------Results-------
Round_Robin                          = Round Robin
Rounds                               = --------Rounds--------
Rounds_to_play                       = Rounds to play
Save_Changes                         = Save Changes
Save_schema                          = Save schema
save_tournament_dat                  = save tournament data
select_an_action_                    = select an action ...
set_import_players_                  = set import players options
set_tournament_options               = set tournament options
SET_UP_TOURNAMENT_DATA               = SET UP TOURNAMENT DATA
Setup_schema                         = Setup schema
Sex                                  = Sex
Swiss                                = Swiss
Team                                 = Team
Team_A                               = Team A
Team_B                               = Team B
Team_Name                            = Team Name
team_ranking                         = team ranking
Teams__Players                       = Teams & Players
Test_schema                          = Test schema
then_press_Tes                       = then press 'Test schema' to see the import result
                                     = Title
```

Do not change left side!

Standard strings

JavaPairing_it.properties

```
Reset                                = Ripristina
result                               = risultato
Results                              = ------Risultati------
Round_Robin                          = gir. all'italiana
Rounds                               = --------Turni--------
Rounds_to_play                       = Turni da giocare
Save_Changes                         = Salva la modifiche
Save_schema                          = Salva lo schema
save_tournament_data                 = salva i dati del torneo
select_an_action_                    = cosa vuoi fare? ...
set_import_players_options           = imposta le opzioni di importazione de      ocatori
set_tournament_options               = personalizza il torneo
SET_UP_T                             = PERSONALIZZA IL TORNEO
                                     = Personalizza lo schema
                                     = Sesso
Swiss                                = Svizzero
Team                                 = Squadra
Team_A                               = Squadra A
Team_B                               = Squadra B
Team_Name                            = Nome della Squadra
team_ranking                         = class. squadre
Teams__Players                       = Squadre e Giocatori
Test_schema                          = Testa lo schema
then_press_Test_schema_to_see_the_import = quindi premi 'Testa lo schema' per vedere il ri
Title                                = Titolo
```

Italian translation

Pack the language file into the .jar

To help development, in the same folder of this file you will find:

- 'i18n' folder
  - info on internationalization of the program
  - the language and country codes to use to name the language file
  - some localized files ('JavaPairing.properties' need to be translated to other languages)
- 'TOURNAMENTS' folder
  - some real tournament files, may be loaded by the program
- 'DATABASES' folder
  - some FIDE, FSI and German Federations filter to import players
  - some plain text and csv databases downloaded from official sites
- REPORTS folder
  - specifications for report to FED generation
  - some report samples
- 'SWISS' folder
  - FIDE rules for swiss pairing systems
- 'AMALFI' folder
  - rules and test examples of Amalfi Rating algorithm
- 'ROUND ROBIN' & 'KEIZER' folders
  - some related documents
- 'TEST' folder
  - some test/simulation tournament files, may be loaded by the program

For downloads: http://javapairing.sourceforge.net

For contact: Eugenio Cervesato (eucerve@tin.it mobile +39 338 5960366)

If you like this program and use it to manage real tournaments, a donation to the "Bobby Fischer" chess club of Cordenons - Italy will be greatly appreciated. Donation is intended to be a self financing method for the standard activities of the chess club and to improve it. As well as, this will stimulate me to continue the software development. Updates, mailing list and quick bug fix are guaranteed to registered users. For bank transfer

A.S.D. Circolo Scacchi Cordenons
c/o B.N.L. Pordenone
IBAN: IT86 N 01005 12500 000000000299
SWIFT: BNLIITRR

Depending on national law, free donation to sports may be tax deducible. The receipt may be requested, please write to the email address.

Enjoy it ! Feedback is highly appreciated!

I wish to thank people helping in some effort
Loris Cancian, Filippo Capizzi, Silvio Cavicchia, Franca Dapiran, Franco De Sio, Paolo Dei Giudici, Christian Krause, Giuseppe Mancini, Georges Marchal, Maurizio Mascheroni, Sergio Pagano, Roberto Ricca, Bruno Rizzuti, Claudio Ruzza, Eric Schiller, Gabriele Stilli, Luigi Voltolini, Federico Zermian, www.soft82.com, www.softpedia.com, www.sourceforge.net. Java® and NetBeans® development teams, Jan Michael Soan for his PrintMe class.

The 'Swiss Dutch' algorithm as implemented in JavaPairing

General rules are:
- the number of rounds to be played is declared beforehand. (Modifying it during the tournament is allowed by JavaPairing …)
- Two players may play each other only once (return tournament may be selected ...)
- White and Black colours should be equal and alternated between games. Three colour sequence or delta colour grater than two are allowed only at the last round for players over 50% if this reduces downfloates. If equal colour due, the strongest colour due takes precedence or the history is evaluated or it is assigned to the better ranked player.
- win scores one point, draw ½ point, loss zero points. Forfeit and BYE score one point and the game is considered played without opponent and colour
- IDs have to be assigned by decreasing Elo (it may be needed to verify the order of ex-equo with other criteria like title, alphabetic, etc ...)
- late entrant and latecomers are handled. Ranking procedure takes them into account dynamically
- Starting from the top of the ranking, a player is tried to be paired within the same score group. 1.st half (S1) is paired with 2.nd half (S2) permuting S2 if game already played or colour or float constrains (not allowed two consecutive floats or as two rounds before, except at the last round for players over 50% if this reduces downfloaters). If impossible to pair, exchenges between S1 and S2 are performed. Unpaired elements are moved to the following group. If gone to the end, the last group is joined with the previous one and the procedure is repeated.
- if odd players, last unpaired, not having had BYE or Forfeit previously, gets the BYE

n.b. The implementation is full compliant to FIDE rules (see Handbook 2010 FIDE)

The 'Swiss Dubov' algorithm as implemented in JavaPairing

General rules are:
- the number of rounds to be played is declared beforehand. (Modifying it during the tournament is allowed by JavaPairing …)
- Two players may play each other only once (return tournament may be selected ...)
- White and Black colours should be equal and alternated between games. Three colour sequence and delta colour grater than two are not allowed. In case of exactly equal colour history, white is assigned to the player with the higher ARO (average opponent rating)
- win scores one point, draw ½ point, loss zero points. Forfeit and BYE score one point and the game is considered played without opponent and colour
- IDs have to be assigned by decreasing Elo
- late entrant and latecomers are handled. Ranking procedure takes them into account dynamically
- if odd players, last ranked, but only one time in the tournament, gets the BYE
- Starting from the top of the ranking, a player is tried to be paired within the same score group. If odd, a player with balancing colour due is floated from the next score group (according to rule 2.5). If needed, colour dues are balanced exchanging some allowed elements of the group, then 1.st half (colour due White) is ordered by increasing ARO, increasing Rating and alphabetically; 2.nd half (colour due Black) is ordered by decreasing Rating, decreasing ARO and alphabetically. Finally 1.st half is paired vs 2.nd half, permuting 2.nd half if needed because of game already played. If impossible to pair, all elements of the group are allowed to be mixed. If impossible to pair, the group is then grown with two more floaters; if gone to the end, the last done pair is broken and the procedure is repeated including all left unpaired.

n.b. The implementation is full compliant to FIDE rules (see Handbook 2010 FIDE)

The 'Swiss Simple' algorithm as implemented in JavaPairing

General rules are:
- the number of rounds to be played is declared beforehand. (Modifying it during the tournament is allowed by JavaPairing …)
- Two players may play each other only once (return tournament may be selected ...)
- White and Black colours should be equal and alternated between games. Three colour sequence and delta colour grater than two are allowed only at the last round. In case of exactly equal colour history, the lower ID alternates the last played colour
- win scores one point, draw ½ point, loss zero points. Forfeit and BYE score one point and the game is considered played without opponent and colour
- IDs are suggested to be assigned by decreasing Elo
- late entrant and latecomers are handled. Ranking procedure takes them into account dynamically
- if odd players, last ranked, but only one time in the tournament, gets the BYE **(*)**
- Starting from the top of the ranking, a player is tried to be paired within the same score group. If odd, a player is floated to the next score group. Finally, 1.st half is paired vs 2.nd half, permuting 2.nd half if needed because of colour criteria, trying to make as many regular color-due pairs as possible first **(*)**, or game already played. If impossible to pair, all elements of the group are allowed to be mixed. If impossible to pair, the group is first grown with two more floaters; if gone to the end, the last done pair is broken and the procedure is repeated including all left unpaired **(*)**

n.b. (*) = this is a simplification of the 'Swiss Based on Rating' algorithm
In the folder 'swiss' you'll find details of 'implementation of the Swiss in JavaPairing'

The 'Swiss Perfect Colours' algorithm as implemented in JavaPairing

General rules are:
- the number of rounds to be played is declared beforehand. (Modifying it during the tournament is allowed by JavaPairing …)
- Two players may play each other only once (return tournament may be selected ...)
- First and last rounds are generated with standard swiss rules
- Except of that, white and black colours are equal and exactly alternated between games. Note that, to achieve this goal the elements of a pair may have different score, especially when many rounds have been played. It is your responsibility to switch to normal swiss if pairs become too much heterogeneous, if you like. The program warns you to switch to regular swiss system when there is only one player remained at the top of the ranking or only two rounds are left to play
- win scores one point, draw ½ point, loss zero points. Forfeit and BYE score one point and the game is considered played without opponent and colour
- IDs are suggested to be assigned by decreasing Elo
- late entrant and latecomers are handled. Ranking procedure takes them into account dynamically
- if odd players, last ranked of the dominating colour, but only one time in the tournament, gets the BYE
- Two groups are created: S1 and S2, each containing half players. S1 has colour due white and is ordered by decreasing score, increasing ARO and increasing Rating; S2 has colour due black and is ordered by decreasing score, decreasing Rating and decreasing ARO. Finally, S1 is tried to be paired vs S2. If some resulting match has been already played, small permutations of the S2 group are performed

The 'Amalfi Rating' algorithm as implemented in JavaPairing

General rules are:
- the number of rounds to be played is declared beforehand. (Modifying it during the tournament is allowed by JavaPairing ...)
- Two players may play each other only once (return tournament may be selected ...)
- White and Black colours should be equal and alternated between games. Three colour sequence and delta colour grater than two are allowed only at the last round. In case of exactly equal colour history, the player with the lower ID alternates the last played colour
- win scores one point, draw ½ point, loss zero points. Forfeit and BYE score one point and the game is considered played without opponent and colour
- IDs are suggested to be assigned by decreasing Elo
- late entrant and latecomers are handled. Ranking procedure takes them into account dynamically
- if odd players, last ranked, but only one time in the tournament, gets the BYE
- Starting from the top of the ranking, a player is tried to be paired with the one that follows it of a number of positions equal to the number of rounds left to play. If the opponent is not legal, a nearer one is first tried and if necessary a farer one. If none legal, the previous calculated pair is broken and a new one is tried.

n.b. To be full compliant to the official rules, some workarounds are needed:
1. If order by Title is needed, just insert players in the right order, or arrange players in the main grid with the aid of right mouse button, or temporarily alter the Elo value before the first round, or externally edit the tournament file.
2. To assign half point to late entrants complete pairings by hand and assign ½-½f

The 'Round Robin' algorithm as implemented in JavaPairing

General rules are:
- the number of rounds to be played is equal to the number of players minus one (double for return)
- players play all against all
- pairing is done according to the official Berger tables
- White and Black colours are best assigned because of the Berger tables. For return the colours of the first match are inverted
- win scores one point, draw ½ point, loss zero points. Forfeit and BYE score one point and the game is considered played without opponent and colour
- IDs are assigned according to the selected setup option
- late entrant are handled, latecomers no! (you should switch to Swiss ...)

JavaPairing has been widely stressed, alone or compared with other similar programs. Here and in following pages you'll find some suggestive test.

Cross Table for Players

ID	Player	Rounds played					score
1	one	0	=W6	=B8	=W10	=B9	=W7	2½
2	two	0	=B7	=W9	=B6	=W8	=B10	2½
3	three	0	=W8	=B10	=W7	=W6	=B9	2½
4	four	0	=B9	=W7	=B8	=W10	=B6	2½
5	five	0	=W10	=B6	=W9	=B7	=W8	2½
6	six	0	=B1	=W5	=W2	=B3	=W4	2½
7	seven	0	=W2	=B4	=B3	=W5	=B1	2½
8	eight	0	=B3	=W1	=W4	=B2	=B5	2½
9	nine	0	=W4	=B2	=B5	=W1	=W3	2½
10	ten	0	=B5	=W3	=B1	=B4	=W2	2½

a.1) 10 players, all games draw, 'Swiss Simple' system

There is always only one homogeneous score group. 'Swiss Simple' strongly apply the rule 'S1 vs S2', so players 1-5 are paired with 6-10. It is curious that if round 6 is requested it stops because at that point there is no solution even mixing all the group!

a.2) 10 players, all games draw, 'Swiss Dutch' system

Starting from 3.rd round it mixes S1 and S2 to obtain perfect colour alternation

b) 24 players, decreasing Elo, 6 rounds, wins always the higher rated

Swiss Dutch

Rank	ID	Player	score	ARO
1	1	A 2600F	6	2400
2	3	C 2520F	5	2366
	2	B 2560F	5	2346
4	6	F 2400F	4	2313
	5	E 2440F	4	2293
	4	D 2480F	4	2286
	7	G 2360F	4	2193
	10	J 2240F	4	2160
9	8	H 2320F	3	2286

Swiss Simple

Rank	ID	Player	score	ARO
1	1	A 2600	6	2406
2	2	B 2560	5	2400
3	3	C 2520	4	2386
	5	E 2440	4	2313
	6	F 2400	4	2300
	4	D 2480	4	2280
	7	G 2360	4	2233
	8	H 2320	4	2233
9	9	I 2280	3	2200

The final ranking should be ID 1-2-3-4-5-6-7-8-9 ... is it surprising that only the first 3 are correctly ranked (by the Swiss Simple engine)?

c) Corus 2010. 14 players/13 rounds round robin, simulated as it was played on 5 or 6 rounds only, by using the real results as virtual results.

The table gives the final real ranking and those obtained with different systems. What do you think is better result? And what about Kramnik performance?

Rank		#	*	^	§	°	##	**	^^	§§	°°
1	Carlsen, Magnus	1	4	3	1	1	1	1	1	1	1
2	Shirov, Alexei	6	5	6	2	5	9	8	5	3	9
2	Kramnik, Vladimir	11	9	11	6	11	8	5	9	11	7
4	Anand, Viswanathan	3	1	5	7	2	2	2	3	4	2
4	Nakamura, Hikaru	2	2	2	3	3	3	4	2	2	3
6	Karjakin, Sergey	6	7	1	4	7	5	9	7	5	6
6	Ivanchuk, Vassily	4	3	4	10	4	4	6	4	6	4
8	Leko, Peter	7	11	12	8	8	7	3	10	9	8
8	Dominguez, Lenier	10	10	8	11	9	6	7	8	10	11
10	Caruana, Fabiano	9	12	7	12	6	11	10	6	8	5
11	Van Wely, Loek	14	14	10	5	14	14	11	14	7	10
11	Short, Nigel	12	13	14	13	13	12	13	13	13	12
13	Tiviakov, Sergei	5	6	13	9	10	10	14	12	14	13
13	Smeets, Jan	13	8	9	14	12	13	12	11	12	14

Legend
= swiss Dutch * = Amalfi Rating ^ = swiss Dubov § = swiss Simple ° = swiss Lim (Vega)

single sign = 5 rounds double sign = 6 rounds

Tips & tricks

a) you may have several tournaments open at the same time running several instances of the program. It is not necessary to save tournaments in different folders. Be only careful that the file 'temp.html' is overwritten each time you get some output. Several files may be merged in memory. A team tournament may be reused as individual. A log dialog helps to understand the pairing procedure (check 'explain' box).

b) if any problem with the printer use the option 'send to browser' or directly open the 'temp.html' file with your preferred browser. This way better pagination is obtained

c) if you need to force the BYE assignment, go back to the 'Registration' page and set temporarily the player 'retired'

d) if you need to exchange colours (maybe player's mistake), go back to the 'Rounds' page, select the pair and by using the right mouse button exchange the colours. Click 'done!' and complete missing results

e) what does it mean 'absolutely free' in the setup window? This option means any previously already played game are not checked. In regular tournament it is a nonsense (good for Title championship?). Maybe useful to organize a team match between two chess clubs with many return games. i.e. Club A 10 players, Club B 10 players. Match are:
Club A - Club B
Club B - Club A
Club A - Club B etc... as many times as needed. Players may be paired 1-1, 2-2, 3-3, etc.. at 1st round; 1-2, 2-3, 3-4, etc... at 2nd round; etc ... do it by hand!

f) if the result of a previous round game is wrong, return to the 'Rounds' page, decrease the round counter, go to the 'Results' page, correct the mistake, click 'done!', go back to the 'Rounds' page and reset the round counter

g) anytime you may remove last round from the 'Rounds' page. If 1st round is also removed, IDs are later newly assigned, so be careful!

h) JavaPairing does not save data timely. It's better for you sometimes to save the work by clicking the 'floppy' icon. Anyway, if any to save a warning is issued when exiting or opening another tournament

i) Output ranking has positions missing. This to remark ex-equo players which are ordered by tie-break criteria. Maybe completed by hand

j) ID assigment does not fullfil requirements. The sorting criteria is selectable in the setup window, default is ‘Elo’. The second official criteria, ‘FIDE Title’ at the moment is not taken into account but some workarounds are in place:
1. enter the players in the correct order or sort the grid by clicking two times on the Elo column header or by using the right button of the mouse (sort descending; move up) to fine tune the list
2. alter the Elo value of a player so that the order becomes correct. Do 1st round and then reset to the original value (in the following rounds ex-equo are ranked by ID which is fixed)
3. (advanced user) do 1st round, save the data and with a text editor exchange the IDs of the players not ordered, save and reopen the file with JavaPairing. Output the players list and the 1st round

k) what is 'acceleration'? In a Swiss tournament it may happen that players are more than 2 elevated the number of rounds (i.e. 40 players/5 rounds). To avoid joint winners, without direct match, the 'acceleration' principle is introduced. The aim is to pair the higher graded (i.e. 'top half') players together, and then to use the top half non-winners to wipe out the 100% scores of 'bottom half' players as quickly as possible. Acceleration ceases when this is achieved or when there is only 1 round left, whichever comes first. The system assumes that lower graded players will not repeatedly beat higher graded players. Like any probability-based system, it can be upset by a sequence of unlikely results. This does not alter the fact that in the long run it is the system most likely to avoid joint winners on 100%.
JavaPairing starts acceleration when it is needed and removes it when the goal has been reached; there are no options in the setup window! Anyway, if you need to force acceleration, temporarily reduce the number of rounds to play. If you want to exclude acceleration, temporarily increment the number of rounds to play

l) Doing a Swiss tournament JavaPairing suggests to switch to Amalfi Rating! It means that, because the low number of players in relation with the round to play, a huge heterogeneous score group is selected, with more than half players! In this situation the usual S1 vs S2 rule alters the tournament and probably leads to unfair secondary prizes win. Amalfi Rating is suggested because it performs pairing in strict ranking order. It you agree, remove the round, go to the setup window and change the play system, then redo the round and compare calculations

m) what about a player not included in the official database? I suggest to treat those players last. You may search in a different database, if any, repeating the standard procedure. If a player is new to Federation you have to insert data directly into the grid on the right of the main window

n) I've a my own database, maybe a pre-registered list, and want to import all players at a step. Yes! It is allowed, but only for individual tournaments (check to have set correct options in the setup ...). First, you have to export your data in .txt or .csv format. If you have a sheet save it in .csv specifying ';' as field separator. If you have a text document containing a table convert it to plain text specifying ';' as field delimiter and none as text delimiter. It is not necessary to remove empty rows, they will be ignored by JavaPairing. Now you can execute the standard procedure to import players in JavaPairing, creating your own filter and testing it. Finally, digit '*' (a single asterisk) in the field 'load player from database' in the main window and do search. If all working fine you'll find all players loaded into the lower grid, at the maximum capacity allowed

o) What does it mean 'tranche FIDE' shown in the Elo variation output page?. This is combined with the 'K' column. Players registered with K=0 are assumed to be Elo FIDE rated; those having K>0 are assumed to be Elo national rated. Both variations for FIDE rated (games played between them, assuming K=30/15/10) and for national rated (all games) are calculated. For national Elo rated the performance against Elo FIDE rated opponents is calculated. Remember, to enter Elo FIDE list you have to complete the required 9 games in two years from the first valid tranche having at least 3 games, 1 point and performace 1200. Then these games are computed together as were played in a single tournament and performance>=1200.

Miscellanea

1. **What does it mean cross-platform, open-source, free-software, GNU GPL, Java?** These are informatics terminology to mean a) cross-platform = the program runs on every operating system (Windows, Linux, MacIntosh, etc.), b) open-source = source code is available to the user, so anyone can see what it really does, c) free-software = the program is for free (a donation is highly appreciated), d) GNU GPL = the license that grants none can impose his own Copyright to the detriment of the author and the development team; it remains forever at everyone disposal, e) Java = the programming language and development environment. JavaPairing accepts all

2. **What engine should be better used?** On the whole the engine should be selected depending on the kind of tournament and on the ratio 'players/rounds to play'.
   a) Swiss Dutch (with acceleration if needed) is the default of FIDE tournaments
   b) Swiss Dubov (with acceleration if needed) is suggested if 8 or more rounds are to play and players have Elo differences in a range of max 200-300, because it optimizes colour allocation but may result slower to obtain a clear final ranking
   c) Swiss Simple (with acceleration if needed) is a basic system in all kind of tournament because it is anyhow 'Swiss based on Rating'.
   d) Amalfi Rating has been tested on the field by the Italian Chess Federation. It demonstrates a good alternative to accelerated swiss engines where Elo differences between players are very high (i.e. in big open), obtaining more balanced pairings at 1st round as well as when acceleration is removed. Nevertheless it is somehow weaker in colour allocation. Top games occur at last rounds, independently of the ratio 'players/rounds to play', but it may happen that 2nd and 3rd do not meet
   e) Round Robin is best suitable when players are few and all can play all

3. **What tie-break criteria should be better used?** On the whole there are some suggestions about engine/tie-break association
   a) Swiss and Amalfi engines require one or more of
   - **Buchholz cut 1, Buchholz**, **Buchholz median** are the sum of score of the opponents reduced by the lowest one, total, reduced by the highest and the lowest ones respectively (calculated according to Kallithea 2009)
   - **Direct Encounter** if all the tied players have met each other, then the sum of points from these encounters is used
   - **Sonneborn-Berger** is 1. for Individual Tournaments the sum of the scores of the opponents a player has defeated and half the scores of the players he has drawn with, 2. for Team Tournaments the score made by each opposing team multiplied by the score made against that team
   - **ARO** (Average Rating of Opponents) is the sum of the ratings of the opponents of a player divided by the number of rounds
   - **TPR** (Tournament Performance Rating) is the ARO incremented or reduced by a value obtained from a table depending on the percentage score of the player
   - number of **games won** - number of **games played with black** - **random**
   b) Round Robin engine require one or more of
   - **Sonneborn-Berger** is 1. for Individual Tournaments the sum of the scores of the opponents a player has defeated and half the scores of the players he has drawn with, 2. for Team Tournaments the score made by each opposing team multiplied by the score made against that team
   - **Direct Encounter** is the sum of points against tied players
   - number of **games won** - number of **games played with black** - **random**
   n.b. JavaPairing applies to tie-break the so called 'FIDE rule' (not played games are counted as a draw against the player himself), except for Buchholz where Kallithea 2009 is applied instead.