

**DFTSimuLab**

*BJStrike*

**Blackjack simulator and strategy analyzer**

**Reference Manual**

**Version 4.0**



## Contents

1. Introduction	3
2. Installation Instructions	6
3. How to run BJSStrike	9
3.1 Graphical User Interface (GUI) User Guide	
3.2 Command line invocation	
4. Practice mode	26
4.1 Practice mode option window	
4.2 Playing window	
4.3 Things to keep in mind in practice mode	
4.4 Templates provided for practice and play mode	
5. Summary of BJSStrike Options	34
6. Simulator reports	41
6.1 Basic simulation reports	
6.2 Optional reports provided	
6.3 Risk of Ruin (ROR) calculation	
6.4 Calculation of optimal bet variation	
7. General commands for simulation control	50
7.1 Deck composition	
7.2 Segment handling	
7.3 Shuffle point declaration and number of rounds to simulate	
7.4 Rules and dealer action	
7.5 Bonus hand declarations	
7.6 Session walk	
8. Count system declarations	67
8.1 Specifying count systems	
8.2 Setting true count conversion accuracy and rounding modes	
8.3 Other count systems	
9. Declaration of betting variation	71
9.1 Basic betting declaration	
9.2 Special bet deviations	
9.3 Advanced bet variation	
9.4 Bankroll based betting ramp	
10. Strategy declaration	80
10.1 Insurance strategy declaration	

10.2	Predefined playing strategies included in BJSrike	
10.3	User customized playing strategy	
10.4	Surrender play	
10.5	Printing the playing strategy	
10.6	Computer perfect playing strategy	
<b>11.</b>	<b>Hand EV analysis and index generation</b>	<b>97</b>
11.1	Hand EV analysis	
11.2	Index generator	
11.3	Score maximizing index generation	
11.4	Index generation based on sampling (automated from the GUI)	
11.5	Manual index generation based on sampling	
<b>12.</b>	<b>Tracking/Steering applications</b>	<b>116</b>
12.1	Ace prediction analysis	
<b>13.</b>	<b>Blackjack Variations</b>	<b>120</b>
13.1	Spanish 21	
13.2	SuperFun 21	
13.3	Joker bonus	
<b>14.</b>	<b>Applications and examples</b>	<b>127</b>
14.1	Red 7 count system	
14.2	Determining optimal strategy for a card counting system	
14.3	Procedure for manual index generation based on sampling using modified strategy files	
<b>15.</b>	<b>Side bets</b>	<b>136</b>
15.1	Royal Match	
15.2	Generic side bet constructs	
15.3	Lucky Ladies	
15.4	Match the Dealer	
<b>16.</b>	<b>General user tips</b>	<b>141</b>
<b>17.</b>	<b>References</b>	<b>142</b>
<b>18.</b>	<b>Technical Support Information</b>	<b>143</b>
<b>19.</b>	<b>Acknowledgements</b>	<b>144</b>
Appendix I	Analysis of ace prediction	
Appendix II	List of setting files and user templates	
Appendix III	Details on installation and platform support	
Appendix IV	Customization of certain GUI entries	

# 1 Introduction

*BJStrike* is a powerful Blackjack simulator and index generator that supports a wide spectrum of counting systems, betting strategies and user customization options. In addition, BJS strike comes with a module for playing Blackjack and practicing counting and betting systems.

BJStrike is useful for determining win rate of card counting systems and to estimate the risk involved in real play. BJS strike comes with a user friendly Graphical User Interface (GUI) for getting started quickly (see Section 3.1). In addition, BJS strike can be run from a shell terminal window or in batch mode for more intensive simulation tasks. Multi-thread simulation, which is supported for some operating systems, can obtain almost a linear performance speed-up with number of processor cores or threads.

Bjstrike operates in four main modes: Standard game simulation; Hand expected value analysis; Index generation; and Playing/card-counting practice mode.

## 1.1 Simulator

- BJS strike supports most of the common rule variations including early and late surrender, any number of decks and up to 6 players. Several common bonus hand and side bet pay-offs can be defined. In addition, the user has full flexibility in defining any deck composition and the shoe can be set in an arbitrary state for simulation and index generation. Cards can be added or removed arbitrarily for simulation and index generation.
- The simulator can keep up to thirteen independent user defined counting systems in true count or running count mode, which can be combined arbitrarily in strategy definitions and betting systems. This means that ace side counts and multi-parameter systems can be easily set up. There are several rounding modes supported for TC conversion.
- The simulator comes with various predefined count systems including Hi-Lo, Zen, KO, Red-7, Hi-Opt and OPP. In addition, the user can specify his own tag point value array for running or true count systems for his own counting systems. BJS strike includes predefined strategies for single and multiple deck basic strategy as well as the Illustrious 18 playing strategy indices. In addition, the user can define arbitrary indices referencing any count system in special strategy files.
- BJS strike has several options for "computer perfect strategy", which means that all possible play options are evaluated dynamically in BJS strike for each playing decision and the play (Hit, Stand, Double or surrender) that maximizes EV is chosen. Computer perfect strategy is useful for determining playing efficiency of a strategy. This option can be used both in regular simulation and in practice mode.
- BJS strike includes a powerful *segment handling* procedure for shuffle tracking and ace prediction analysis in which the shoe can be partitioned into segments. Each segment can be shuffled individually, not at all; or merged with other segments at shuffle time. The initial deck state and card order can be fully customized and segments of certain ace or face card

density can be auto-generated at shuffle time and the simulator can be set up to automatically insert ace or face cards into a segment at a given probability.

- The user strategy definition module allows arbitrary composition dependent and penetration dependent strategies to be defined either from scratch or added incrementally on top of standard strategy templates. For example, up to three indices can be defined for each hand composition from any of the 13 count systems. Different strategies can be defined for top of the deck hands or at certain round numbers or based on a certain number of cards drawn. For advanced strategy declarations, an *'if-then-else'* construction is provided. Different betting schemes can be defined for different players or hands for evaluating multiple hand play. Statistics are reported per player as well as in total numbers to evaluate multiple hand strategies.
  
- The standard statistics reports generated include EV, SCORE, standard deviation and hourly win rate. In addition, customized statistics can be generated per count or round number. Betting or sitting hands out can be controlled per round or player based count or win/loss results. Statistics are generated per hand as well as combined for all hands defined as the counter's hands.
- BJStrike can generate Risk of Ruin (ROR) tables as well as calculate optimized betting ramps based on Kelly criterion for any supported strategy. Moreover, there is a special mode, *Session Walk*, in which short-term session or trip ROR can be estimated and effects of variance on session results can be studied.
- Performance on an AMD Duron processor at 1.0 GHz is on average 50 million hands per minute. Index generation mode runs on average at 200 million hands per minute. Unlimited number of hands can be simulated as BJStrike uses full 64 bit arithmetic for all internal counters.
- A fixed bankroll can be specified to estimate risk of ruin through simulation and to estimate number of hands it takes to double the bankroll. A variety of betting schemes are provided that can be mixed with count based betting strategy. A *playback* mode is provided in which each hand played is reported in the results window including cards dealt, betting, playing decision, payoff and count. The shuffle procedure can be switched off and the deck stacked to evaluate certain strategies of user controlled shuffle.
- BJStrike supports a variety of rule variations including Spanish 21 with redouble in simulation and regular index generation
- BJStrike can analyze common bonus hands and side bets including Royal Match, Lucky Ladies and Match the Dealer bets and associated suit based count systems. In addition, there are generic constructs for specifying sets of arbitrary bonus and side bet hands based on the dealer and player first 2 cards.

## 1.2 Index generator and playing hand analyzer

The index generators provided in BJSStrike are simulation based, which means that the indices are generated by simulating the two plays independently and determining the EV for a range of consecutive counts. Next, the count at which the EV functions of the two plays meet is reported. Realistic representative deck compositions for each count are generated by drawing cards randomly from a shuffled deck under certain criteria. Many different deck compositions are evaluated for each count. The distribution of deck compositions evaluated for a certain count will follow the normal distribution of compositions that occur in normal play, which means that the indices can be accurately generated under the same playing conditions as they are used.

- Index generation can be run in either EV maximizing (normal) or SCORE maximizing (commonly called risk-averse) modes. The purpose of SCORE maximizing mode is to sacrifice some EV in a strategy at the benefit of lower variance or risk, and, hence increase the SCORE.
- The deck can be set in an arbitrary state for index generation and there are options to specify at which penetration to generate the indices. Both running and true count systems are handled.
- The expected value vs. count can be reported during index generation. Composition dependent indices can be generated for up to 6 card hands as well as total (composition independent) indices.
- A hand analysis mode is provided for estimating EV of a certain hand composition vs. dealer upcard and for basic strategy generation for arbitrary deck composition and penetration.

## 1.3 Practice mode

- Play one or multiple (1-5) hands at a blackjack table or just watch a game being played by all computer players for practicing back-counting. Auto play mode can be used for realistic card counting drills. The user interaction for betting and playing decision during play is handled through the standard mouse buttons or certain dedicated keys to provide a comfortable feeling and fast playing speed.
- Most of the rule variations and set-ups available for simulation are also available in playing mode including advanced betting and playing strategy control and multiple card counting systems. In addition, computer perfect playing strategy can be used in practice mode with any deviation flagged and reported in the results file.
- Action on playing decision error or betting error can be set to auto-correct and notify or simply notify. Statistics about number of playing and betting errors can easily be checked at each round. A window showing running count, true count, penetration as well as suggested play according to active strategy can be brought up at each round.
- The animations can be fully customized using 20 different delay parameters to accurately emulate specific playing conditions or choose from several pre-customized speed options.

## 2 Installation Instructions

All simulator versions come as compressed tar file archives (zip or dmg) in the format `<sim_type-ver.platform>.tar.gz` or `<sim_type-ver.platform>.zip`. The file name of the compressed file indicates simulator options, release version and system OS platform type. The compressed tar file is less than 1 Mbytes in file size.

`<sim_type>` is `bjstrike_D`, `bjstrike_S` and `bjstrike_X` for the demo, limited and full edition versions, respectively.

Notice, any old version of BJSrike (demo or full version) needs to be uninstalled prior to a new installation.

### 2.1 Installation instructions

Save your compressed distribution file into your desired installation directory.

#### 2.1.1 Windows operating systems

You will first need to install the tcl/tk graphical toolkit and then you start BJSrike by double clicking on the `bjst.tcl` icon in the unzipped folder. Without the tcl/tk toolkit neither the graphical user interface nor the uninstall application will work.

##### 1. Installing the tcl/tk toolkit

Go to the link at:

<http://www.activestate.com/Products/ActiveTcl/> or

[http://www.activestate.com/activetcl/tcl8\\_6/](http://www.activestate.com/activetcl/tcl8_6/) to download the latest version of "ActiveTcl"

This is a free open source graphical toolkit which is available for all operating systems but not pre-installed in Windows. Version 8.3.4 or later of the tcl/tk toolkit is required for BJSrike.

Next, you execute this file and choose the default options for the few questions that will appear.

##### 2. Starting bjstrike

Any older version of BJSrike including demo versions must be uninstalled prior to installing a new version. Choose "Uninstall BJSrike" from the 'Environment' menu of the main window to uninstall an old version.

Unzip the `bjstrike` zip file and start the simulator by double clicking on the 'bjst' (or `bjst.tcl`) icon. Do not start `bjstrike.exe`. This is the core simulator binary, which gets called from the user interface (`bjst`).

The toolkit in (1) should automatically get invoked when `bjst` is started. The very first time you start `bjst`, it should ask if you want to register. Answer yes if the terms are accepted and then enter the registration PIN code in the License information window that will show up. For demo versions any number can be entered, for example 123. Notice, for full versions the attached PIN code is valid for one calendar month only. A new PIN can be obtained from the support web page if the software is installed later.

(See: <http://www.dftsimulab.com/bjstrike/support/main.html>)

Next, the simulator will ask for a 'workspace' folder. Use the default suggestion here. You are now ready to run simulations.

### 3. Testing

For testing the simulator select the "SIMULATION MENU" button from the main window. In a second window that shows up press the "Start simulation" button at the bottom of the window and a short simulation will run. To test the blackjack practice mode, select the "PRACTICE" button from the main window. In the second window that shows up, press the "Start game" button. Notice that for demo versions, only card ranks 2,5,9,T exist.

If an incorrect PIN code was entered during the registration the simulator will give an error message when you start a simulation and you will need to uninstall bjstrike using the 'uninstall' application or from the 'Environment' menu of the main window before a new PIN can be tried.

## **2.1.2 Linux/Unix (Solaris)**

### 1. Graphical toolkit

First verify that you have the tcl/tk graphical toolkit installed on your system. For checking your system see: [http://www.dftsimulab.com/bjstrike/support/install\\_tcl.html](http://www.dftsimulab.com/bjstrike/support/install_tcl.html)

The environment in which the GUI (bjst) can function requires that, in a shell terminal window, typing the command:

```
echo 'puts $tk_version; exit 0' | wish
results in a value of 8.3 or greater printed.
```

Most Linux distributions come with this toolkit pre-installed. This is a free open source graphical toolkit which is available for all operating systems. If you don't have the toolkit installed go to:

<http://www.activestate.com/Products/ActiveTcl/> or

[http://www.activestate.com/activetcl/tcl8\\_6/](http://www.activestate.com/activetcl/tcl8_6/) to download the latest version of "ActiveTcl".

Version 8.3.4 or later of the tcl/tk toolkit is required for BJStrike.

### 2. Downloading and unpacking the BJStrike package

Download the distribution file (note: for full versions the distribution package is attached in the order cover e-mail) into your desired installation directory and from a terminal window run:

```
% gzip -d {File}.tar.gz
% tar xvf {File}.tar
% cd {File}
% ./bjst & (Start of BJstrike)
```

### 3. Registration

The very first time you start bjst, it should ask if you want to register. Answer yes if the terms are accepted and then enter the registration PIN code in the License information window that will show up. For demo versions any number can be entered, for example 123. Notice, for full versions the attached PIN code is valid for one calendar month only. A new PIN can be obtained from the support web page if the software is installed later. (See: <http://www.dftsimulab.com/bjstrike/support/main.html>) Next, the simulator will ask for a 'workspace' folder. Use the default suggestion here. Now, you are ready to run simulations.

## 4. Testing

For testing the simulator select the "SIMULATION MENU" button from the main window. In a second window that shows up press the "Start simulation" button at the bottom of the window and a short simulation will run. To test the blackjack practice mode, select the "PRACTICE" button from the main window. In the second window that shows up, press the "Start game" button. Notice that for demo versions, only card ranks 2,5,9,T exist.

If an incorrect PIN code was entered during the registration the simulator will give an error message when you start a simulation and you will need to uninstall bjstrike using the 'uninstall' option from the 'Environment' menu of the main window before a new PIN can be tried.

### **2.1.3 Mac OS**

#### 1. Distribution file

The distribution file for Mac is a compressed self-contained disk image (.dmg) named {sim\_type-ver.platform}.dmg. This application is of type 'bundle' and already contains the tcl/tk toolkit needed for the user interface embedded in the application. Thus, there is no need to install any additional software on Mac in order to run bjstrike. Copy this file to a local user area and double click on the application icon 'bjst' in the main folder.

#### 2. Registration

The very first time you start bjst, it should ask if you want to register. Answer yes if the terms are accepted and then enter the registration PIN code in the License information window that will show up. For demo versions any number can be entered, for example 123. Notice, for full versions the attached PIN code is valid for one calendar month only. A new PIN can be obtained from the support web page if the software is installed later. (See: <http://www.dftsimulab.com/bjstrike/support/main.html>) Next, the simulator will ask for a 'workspace' folder. Use the default suggestion here. Now, you are ready to run simulations.

#### 3. Testing the installation

For testing the simulator select the "SIMULATION MENU" button from the main window. In a second window that shows up press the "Start simulation" button at the bottom of the window and a short simulation will run. To test the blackjack practice mode, select the "PRACTICE" button from the main window. In the second window that shows up, press the "Start game" button. Notice that for demo versions, only card ranks 2,5,9,T exist.

If an incorrect PIN code was entered during the registration the simulator will give an error message when you start a simulation and you will need to uninstall bjstrike using the 'Uninstall' application, that is found under the 'Environment' menu of the main window, before a new PIN can be tried.

## **2.2 Support**

Updated new versions of the reference manual can be ordered from the *BJStrike* support section of DFTSimuLab web site at [www.dftsimulab.com/bjstrike/](http://www.dftsimulab.com/bjstrike/)

For support go to [www.dftsimulab.com/bjstrike/](http://www.dftsimulab.com/bjstrike/) and select the link *Technical Support* from the main page or send e-mail to [support@dftsimulab.com](mailto:support@dftsimulab.com).

## 3 How to run BJSStrike

BJStrike can be run either using the Graphical User Interface, *bjst* (Section 3.1) or in command line mode from a shell terminal window (Section 3.2).

### 3.1 Graphical User Interface (GUI) User Guide

BJStrike comes with a GUI that provides an easy way of learning and getting started using the simulator or practicing Blackjack play. For a guide on how to use the *Play and Practice* option go to Section 4.

The GUI requires the Tcl/Tk graphical toolkit of version 8.3.4 or later to be present on your system. Tcl/tk is a standard open source package, which is distributed for free to most of the common system platforms including Linux/Unix, Mac OS X and Windows. Most Linux distributions come with this library pre-installed as standard and there are binary packages available for easy installation using standard software installation procedures.

The GUI is started by double clicking on the icon *bjst* in the application folder or from a terminal shell by typing

```
<Installation directory>/bjst
```

where *<Installation directory>* is the path to the installation folder.

The very first time the GUI is started, the software needs to be registered and a workspace folder need to be specified as described in Section 3.1.1. This configuration only needs to be done once per installation as the environment information is stored in the user's home directory under the file name ".bjstrikerc" (*xbjstrike.rc* for Windows) and reloaded each time *bjstrike* is started.

Section 5 lists the GUI options available for each simulator command including references to what GUI windows the option is located in.

#### 3.1.1 Registration and setting up BJSStrike user environment

There are two items associated with *bjstrike*'s environment that need to be set up in the GUI to provide smooth usage. This configuration only needs to be done once.

- (1) Start the GUI: Double click on the *bjst* icon or from a command line terminal type: *<Installation dir>/bjst*  
The main window (shown in Figure 1) will show up.
- (2) The first time the GUI is started a license acceptance window will show up, which asks for acceptance of the User License Agreement and for the user to enter a Registration PIN code. The one-time personal PIN code is included in the Cover letter of the software distribution and is valid for one calendar month. If the installation needs to be done later or on a new computer a new PIN can be automatically generated at DFTSimuLab's support web site.  
If an incorrect or expired pin code is entered the simulator will abort when a simulation/index generation is launched. The software package has to be uninstalled

from the menu, *Environment->Uninstall* of the main window or using the “*uninstall.tcl*” application and then re-started from the distribution folder before a new registration can be attempted.

- (3) After entering the registration PIN, the simulator will suggest a default system specific temporary directory to be used as workspace. Accept this default suggestion to start with. If this folder later needs to be changed select “*Set Workspace*” from the *Environment* menu and choose a folder to be used as workspace directory using the file hierarchy browser that shows up. The workspace is simply a directory in which the simulator is executed and where the results files and temporary run-time files are generated. This directory can be any directory in which the user has write permission. It is recommended to use another directory than the installation directory as workspace<sup>1</sup>.

Other environment variables, “*BINDIR*” and “*LIC. FILE*” are identified automatically during the registration and should never need to be touched by the user. If not, they can be changed manually from the *Environment* menu.

The environment setting can be inspected or changed any time from the *Environment* menu of the main window by selecting “*View environment*” or “*Set Environment*”.

It is required to uninstall older versions of BJSrike including demo versions using the menu *Environment->Uninstall* of the main window or using the GUI application *uninstall.tcl* before a new version is installed to properly clean up all registration and configuration files. Conflicts may otherwise occur between new and old versions and registration numbers, which may cause failure in starting up the new version.

### 3.1.2 Basics of using the GUI

In general, for text field input options, leaving the entry empty will have the effect of selecting the default action for that input. The default action chosen by the simulator for the various options can, after a simulation, be found in the result report file or can be looked up in Section 5 of this manual. The result report file can be viewed by pressing the “*View results*” button.

During normal use, after a simulation task is completed, it is recommended to save and load the settings and user customization in files in user directories using the “*Save setting as*” or “*Save setting*” from the *File* menu. When BJSrike is restarted, selecting the “*Load setting*” option and specifying a file name, immediately brings back the customization from a stored session. In addition to the simulator settings, the simulation results are saved and linked to the name of the file name that contains the setting. This means that when a setting is reloaded, the results of the simulation most recently run prior to the save will be brought back and can be inspected immediately using the “*View results*” button. When a new simulation is run and the setting is saved using “*Save setting*” any previous results stored under that name will be overwritten.

BJStrike comes with a comprehensive set of GUI examples and templates for different simulation tasks, which can be loaded and run immediately as described in Section 3.1.8. Appendix II contains a complete list of templates included. Detailed descriptions with comments for some common applications can be found in Section 14.

---

1. If multiple simultaneous sessions of BJSrike is desired, each session has to have a unique workspace defined.

When `bjst` is launched, the main window, shown in Figure 1, appears. There are four main modes in which BJS strike operates: *Regular simulation*; *Hand analysis*; *Index generation*; and *Practice and play* mode. Each mode has a dedicated button at the bottom of the main window. The Practice and play mode is described separately in Section 4.

The main window contains most of the options that are common for all modes of operation including count system declaration, rules and deck state configuration. Selecting a mode of operation brings up a second window, referred to as *application window*, which contains the options that are specific for that particular mode of operation. As an example, Figure 2 and Figure 6 show the simulation and index generation windows, respectively. The bottom bar contains a start button for starting a simulation. Most of the options are self-explanatory. For details about underlying assumptions and procedures, the options are described in details in Sections 6-11 and a summary of all command line options are given in Section 5.

The first times the simulator is used on a new system, it's a good idea to select a fairly small number of rounds to get acquainted with the simulation speed on a particular host system and to learn what run times to expect for the different simulation tasks. The performance is to a large extent dependent on the CPU clock frequency of the host system. Since the active memory workspace is very small during a simulation, the primary memory size of a system has almost no impact on the performance.

The results of the latest simulation or analysis can always be observed at any time by pressing the “*View results*” button from any application window or from the *View* menu of the menu bar from any window. Notice, that it is up to the user to save the results of a task by either using the “*Save as*” option or by manually copying and pasting the relevant text from the results window into his own application or word processor. Any previous unsaved results gets overwritten when a new simulation is launched. The “*View->Environment*” shows the name of the currently active simulation session. On Windows platforms, text from the result window can be copied and pasted into other applications using the standard *Ctrl-c* and *Ctrl-v* key strokes. On Unix/Linux platforms, text can be selected and pasted simply using the left and middle mouse buttons as in standard X window applications.

### **- Main window**

The main window, shown in Figure 1, contains options that are common for all modes of operation including options for specifying deck composition using either the *Composition* or *State* buttons as well as defining customized or standard count systems. The *State* option is used if a certain order of the cards in the deck is desired. This mode is typically used with the shuffle procedure off in playback mode to evaluate or test certain strategies. To enter point values for user customized count systems, first select either *Custom running* or *Custom true* from the menu of any count series. Then, select the tag by clicking on the tag in the tag assignment list box. Next enter the new value and press enter. Rounding mode and true count conversion accuracy can be set in this window. Notice that, although there are only three simultaneous count systems available on this window (C1-C3), BJS strike supports up to 13 systems (C1-C13) in running or true count, which can be defined using custom commands or in a strategy file. This means that a variety of combinations of different count and side count systems can be used in the same session. The shuffle procedure can be switched on and off using the *Rnd. Shuffle ON/OFF* buttons. In default mode an automatically generated number is used as seed for the Random Generator used in the shuffle procedure. A fixed random generator seed can be specified if repeatable results are desired by un-checking the button *Auto random seed*. A good seed to use is 6000011, which is the smallest prime number that is greater than 6000000.

Among the rule variations, RSA stands for Resplit Aces and HSA stands for Hit on Split Aces. The resplit option refers to resplit up to maximum  $n$  hands.

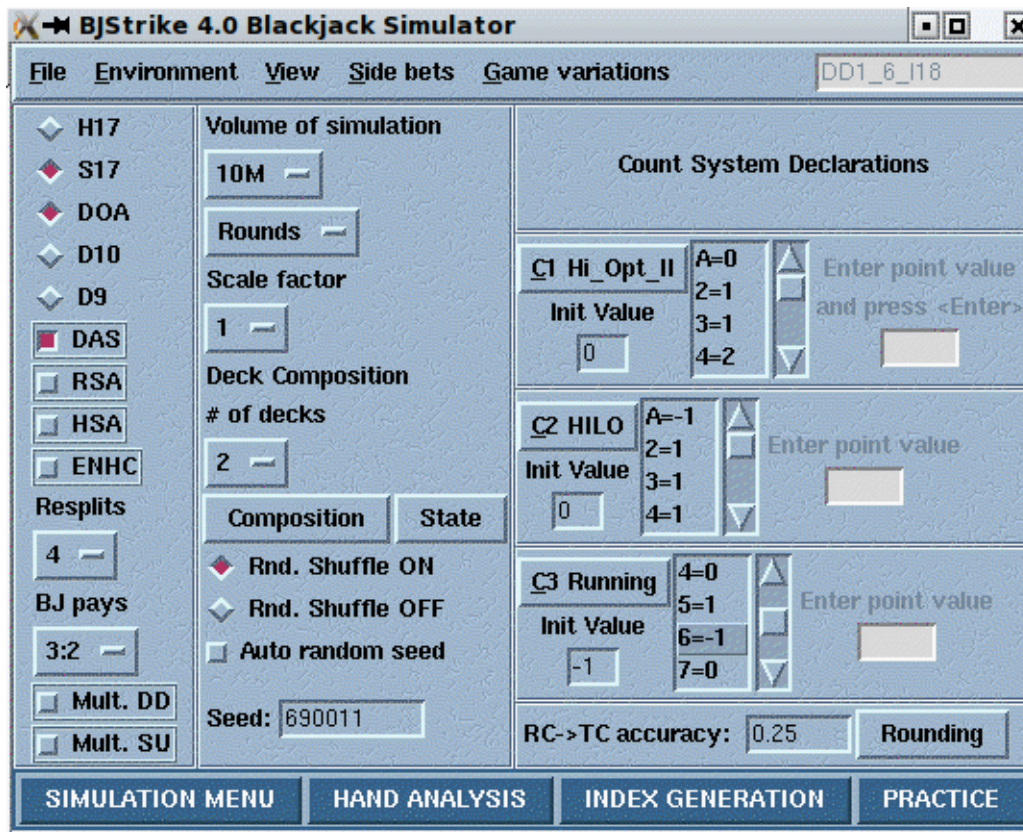


Figure 1 BJStrike main window

### 3.1.3 Simulation window

In the simulation window, shown in Figure 2, betting systems and playing strategy can be defined. The default mode is flat bet of one unit. There are several ways to declare bet variations. Different betting systems based on count can be specified for player 1-3 or the same system can be applied to all players. To enter a betting declaration select player (or all) and a count system, C1-C3, and fill in the count range and betting amount in units. Select an object from the list box or enter new values for count range and amount to bet and press *Add*. An entry in the list can be deleted by first selecting the object and then pressing *Delete*. A count range is defined as  $[C_L, C_U]$  where  $C_L \leq C_U$ . The notation ‘-’ in the left(right)-hand side indicates  $-\infty$  ( $+\infty$ ). The bet amount will be applied for the count  $C$  in the interval,  $C_L \leq C \leq C_U$ . In this mode, any range that is not covered will be assigned the bet zero. In addition there are some options for deviation from the bet variation including: keep same bet after a tie; special top of deck bet; and no reduction in bet amount after a win. Checking any of these options will override the bet amount determined by the count. Notice, that in the list of betting intervals, the entire range of possible count values needs to be covered by the intervals specified. In addition, overlapping intervals must not exist since they can cause conflicting betting amount for a given count leading to incorrect results.

In addition, a more flexible bet entry form, shown in Figure 3, can be opened by pressing the *Matrix* button. This form allows specification of additional bet variation based on various criteria and provides customized betting variation for each player and round number. In addition, options for sitting out hands can be specified for each entry. The check boxes in the first column can be used to select which players should be included in the results reported under *Total*, which is useful for multiple hand play. For any given entry in the matrix, *Inactive* means that the default betting scheme defined in the Simulation window will be applied. To inspect the betting options for an active entry, first press the *Active* button and re-select *Active* from the pull-down menu. The options are then copied over onto the form below the matrix. For an entry that is *Active*, the betting conditions are evaluated from item 1 down to Item 5. The first condition that is satisfied will determine the bet amount. If no active condition is satisfied for a certain round the default betting as defined in the second line of this frame will be applied. “*Use basic bet*” means that the bet specification in the main simulation window will be applied. Conditions for sitting out a hand can be specified in the last line titled “## Play hand .... if” of this frame. Another option in the betting matrix window, “*Re-shuffle on wong out*” can be checked if a re-shuffle is desired after each wong-out of a shoe. The unchecked (default) behavior is to allow jumping in and out of the same shoe solely based on the conditions in the “## Play hand .... if”. In play all mode, this option has no impact (See also Section 9.3). If different Wong-in and Wong-out indices are desired use the WONG\_OUT command described in Section 9.3.1.

Notice, that there must always be at least one hand (player) that plays all hands in wong-in/wong-out simulation sessions. Otherwise, the game cannot progress. For example, while playing two hands with an entry point of TC=0, the simulation should be set up with at least one additional player playing all hands that is not part of the hands selected with the “*My hands*” check boxes. Only the two hands that are specified to conditionally sit out rounds should be selected under “*My hands*”.

If entries for column 0 or row 0 are specified the conditions will apply to all players or all rounds, respectively. If there are active entries for both “all rounds/players” and for specific rounds /players the specific entry will override the all rounds/players entry. A betting declaration for a certain matrix entry is specified by first selecting the matrix entry into active mode. The column and row number should be updated at the top of the entry form and the current setting for this entry will be shown. Then, the desired option fields are filled out and the *Apply* button pressed. If the *Apply* button is not pressed before selecting another matrix entry or closing the window the entries will have no effect. The bet matrix window must be closed before any other windows can be accessed. In order for the bet matrix entries to have effect in a simulation, the *Matrix On* button must be checked in the simulation window. Details about the bet matrix entry constructs can be found in Section 9.3.

The shuffle point can be expressed in number of cards played or number of rounds played.

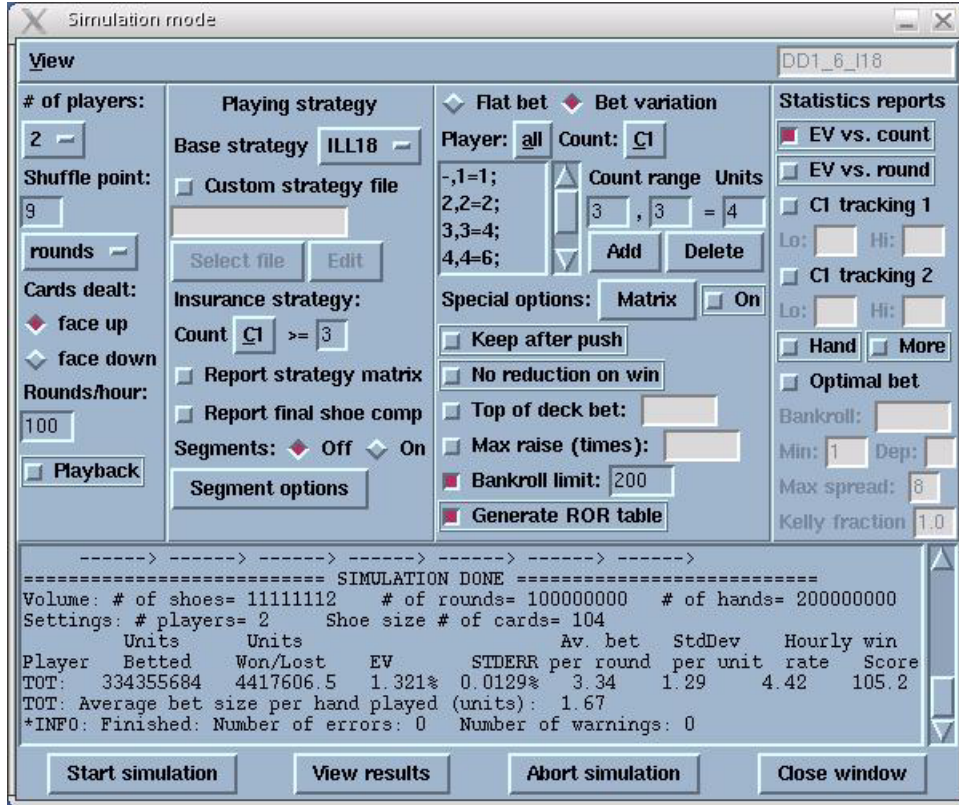


Figure 2 Simulation window, BJStrike version 3.0

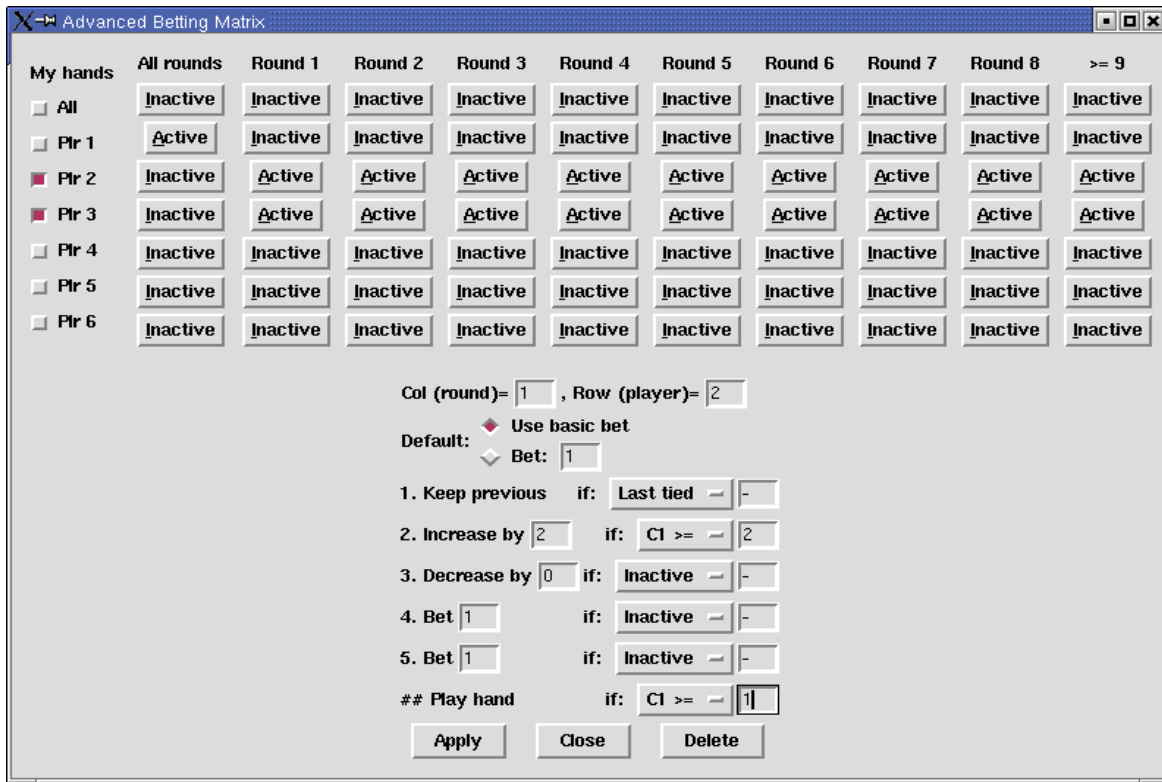


Figure 3 Advanced betting declaration matrix

As an example, a GUI setup for a 6 deck game with 2 hands played and the second hand is wonged out if the count is less than 1 can be obtained by loading file:

```
<Installation directory>/test_cases/GUI_Appl/Wong_6D_TC1
```

To illustrate the results generated in regular simulation, Figure 5 shows a subset of the basic statistics generated in this example.

An example of bet variation vs. round number and player for single deck can be found in file:

```
<Installation directory>/test_cases/GUI_Appl/SD_Depth
```

Furthermore, Section 14.1 presents another example of using wong-in/wong-out when playing two hands.

### 3.1.4 Miscellaneous options

At the bottom of the left column of the Simulation window, a check box *Playback* exists. This option causes each hand played including cards dealt, playing decision and amount lost/won to be written in the log file (See also Section 6.2). In this mode, the number of rounds possible to simulate is limited since the file size of the result file will grow significantly.

In general, to minimize simulation run time, it is recommended to only use the reports or count systems that are needed in a certain simulation session. Each additional report or count or betting system adds computation time, and therefore slows down the overall simulation performance.

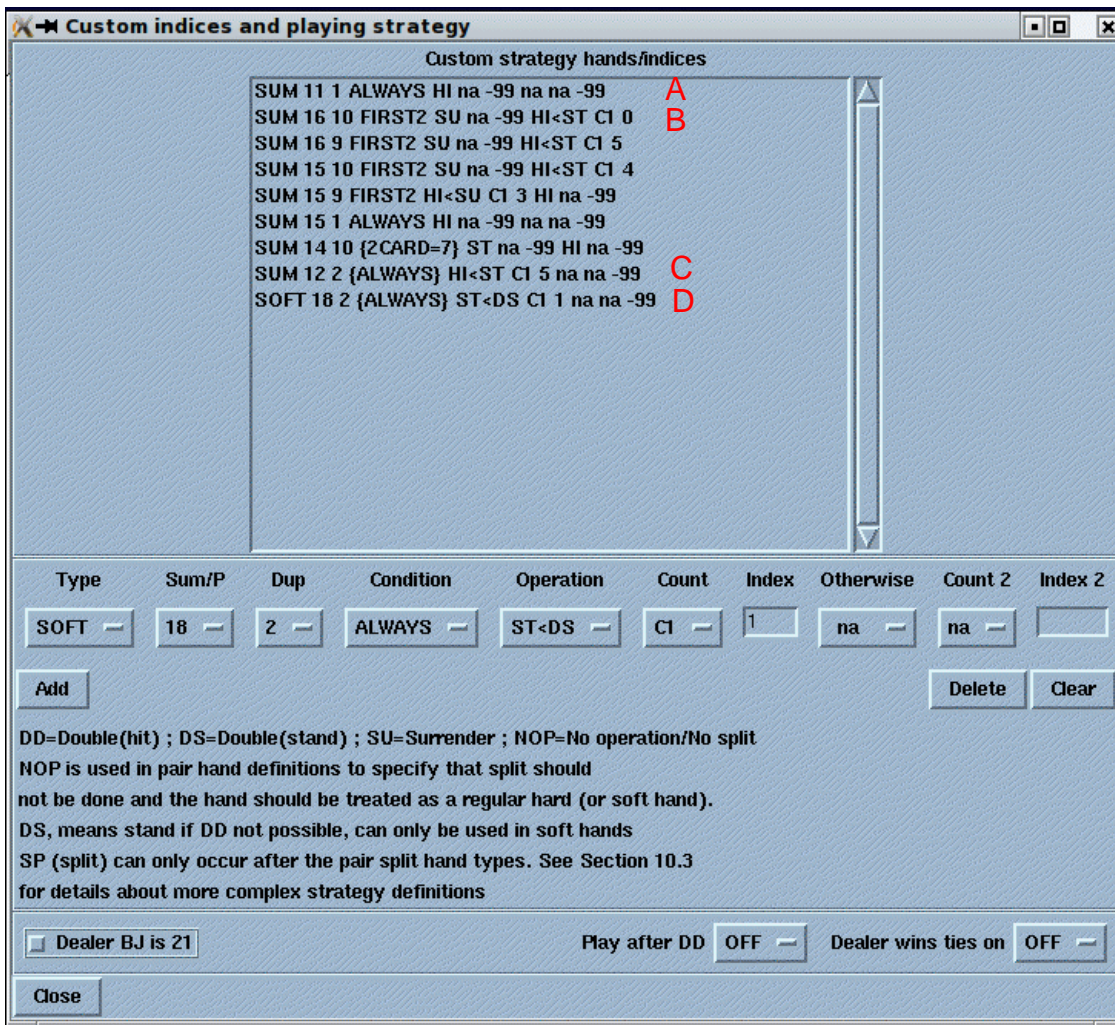
The option window *Bonus hands* can be used to specify some common bonus hand payoffs. There is a much more generic way of defining custom bonus hands in strategy files as described in Section 7.5. Arbitrary user hands and dealer fist two card combinations with variable payoffs with and without auto-win can be specified. Hard, soft and double down hands can be distinguished from one another in specifying bonus hands. In addition, several common side bets can be selected from the *Main window* -> *Side bets* menu options. More generic side bets can be specified using custom side bet commands (see Section 15.2)

### Playing strategy specification

In addition to the predefined playing strategies selected under *Base strategy*, user custom strategies and indices for any number of individual hands can be entered by opening the “*Custom Strat. entries*” form from the simulation window. Figure 4 shows an example of user custom strategies specified on the Custom strategy form. Custom strategy specifications and indices always override the predefined base strategies. For a given playing strategy system only the hands for which a change from the *Base strategy* is desired need to be specified. One level of conditional strategy expressions can be entered. If more levels and conditions are needed the strategy needs to be specified in a strategy file. The entry “*Condition*” on the custom strategy form selects under what condition, for example always or in a 2-card hand, the playing strategy or index under “*Operation*” should apply. If it is an index the count series and index number is entered in the subsequent entries. Next, if the condition is not “always” another playing decision or index must be specified under “*Otherwise*”.

In addition, custom strategy declarations can be written in a strategy file that will override the predefined playing strategy for any hand composition or indices. The constructs available for defining a custom strategy including indices described in Section 10.3 are fairly generic to cover a variety of possible strategy functions. In addition to playing strategy constructs, if needed, any

custom command described in Section 5 can be entered directly in the strategy file (see Section 3.1.12). The *Report Strategy Matrix* option prints the complete active playing strategy in the report file in an easy-to-read format, which is useful when custom strategies are used (see Figure 22 in 10.5).



- A) Specifies hit for hard 11 vs dealer ace
- B) Specifies hard 16 vs T strategy: For a non-splitted two card hand do surrender  
For a multi-card hand or surrender not possible stand if count C1 is greater than or equal to 0.
- C) Specifies index +5 in C1 for hit vs. stand for hard 12 vs. 2.
- D) Specifies index +1for stand vs. double for hard soft 18 (A7) vs 2. If double is not possible stand.

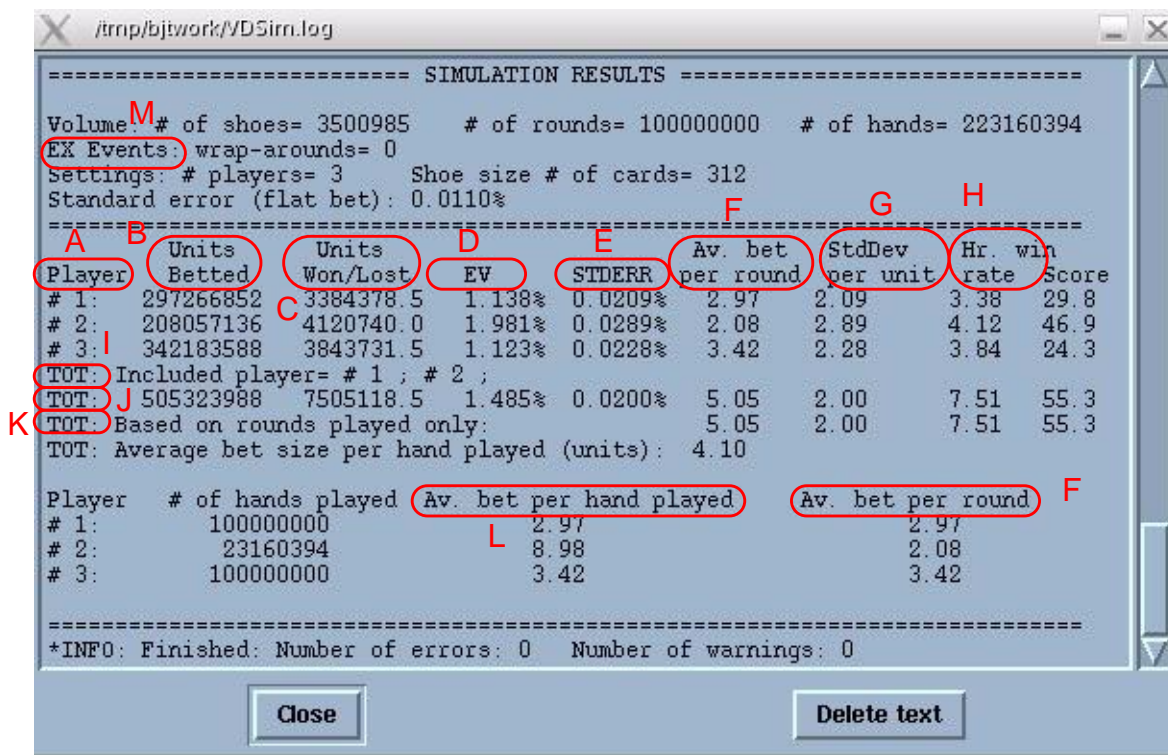
**Figure 4** Custom strategy form for entering playing strategy and indices

The *Segment option* window can be used for tracking and sequencing analyses and the *Computer optimal play* option is used to evaluate a system's efficiency against computer perfect play. Computer optimal play means that dynamically, for each decision, the two candidate plays are simulated and the playing decision with highest EV for that particular deck composition is dynamically chosen. Any difference between the play predicted by a count system and the optimal play is logged and a summary of deviations is generated.

### 3.1.5 Session Walk

From the simulation window the *Session Walk* window can be opened. This mode is useful for analyzing short term trip or session results and implications of variance including maximum number of consecutive rounds lost or won. Furthermore, if a session bank is specified, session or trip Risk of Ruin will be estimated through simulation. If this mode is desired make sure that the *Session Walk Active* check box is marked. In this mode a certain number of sessions are simulated with each session set to a fixed number of rounds. If the "*Report each session result*" button is checked the result of each session will be reported in the results file. Make sure to set the number of session to reasonable small number to avoid big file size of the results file. This mode allows user customized post-processing of session results if so desired. See Section 7.6 for details on Session Walk results. An example of setup for Session Walk analysis can be found in:

```
<Installation directory>/test_cases/GUI_Appl/SessionW1.rc
```



Example of a 3 hand simulation where the player analyzed plays hands #1 and #2 with a wong-in/wong-out index of +1 for hand #2 and play-all for hand #3. This set-up is available at:

<Installation directory>/test\_cases/GUI\_Appl/Wong\_6D\_TC1

Summary of the statistics: (Details and mathematical definitions of all quantities can be found in Section 6)

- A) Table player spot. In multiple hand play, several spots can be combined for the Total results below.
- B) Total amount initially betted for this table spot (in betting units)
- C) Total amount won (including all splits, doubles) (in betting units)
- D) Expected value in %. The EV reported here is IBA (Initial betting advantage), i.e.  

$$\frac{\text{Tot. amount won (C)}}{\text{Tot. amount initially betted (not including doubles and splits) (B)}}$$
- E) Standard Error of the EV percentage estimate.
- F) Average bet per round. This is expressed per round simulated regardless of whether a round is played  
 In this example we have significantly lower average bet for spot 2 since this spot wongs in and out at TC=+1
- G) Standard deviation per unit. To obtain the standard deviation in betting units of a session of N rounds, multiply G by F by the square root of N.
- H) Hourly win rate. This column is EV times average bet times <Number of hands per hour input parameter>
- I) Indicates which table spots are included in the total results (J), "My hands". This is a user input parameter
- J) Accumulative total results of the hands included in "My hands" (J) per round simulated.
- K) Accumulative total results of the hands included in "My hands" (J) per round for which at least one on the hands defined in (I) was played
- L) Average bet per player spot per hand played, this is different from F if a spot is sitting out rounds
- M) Indicates the number of situations for which the dealer ran out of cards, this field should be zero.  
 If not zero, adjust the shuffle point (number of cards drawn or rounds played)  
 Other special events such as number of bonus hand hits and number of special strategy decisions are also reported here.

**Figure 5** Example of results reported in the results file from a regular simulation

### 3.1.6 Aborting a running simulation

A control button at the bottom of any application window can be used to abort any running simulation or index generation. Using the abort button only kills the simulation process while keeping the GUI running and a new simulation task can immediately be restarted.

For most platforms, `<Ctrl-q>` or `<Ctrl-v>` can always be used to abort the GUI and any potentially running simulation. This is a hard abort and no file pointers or current settings will be saved. In order for this key press to have effect, any of the four main windows must be active by the window manager and the mouse must be positioned above the active window.

### 3.1.7 Some words about simulation accuracy

Notice that the number of rounds or hands to simulate is an important parameter related to the accuracy of an analysis as the *standard error* of a simulation is directly dependent on the number of rounds. Standard error is an important measure of the range for which the outcome of a random process simulation may fall with a certain probability. The standard error is reported in most of the statistics reports. The definition of standard error can be found in Section 6.1.1. Per default, the number of rounds is set fairly low for demo purpose and should be increased as the need for accuracy dictates for real applications. In most cases, the requirements for number of rounds for index generation is significantly lower than the number needed for regular game simulation. An example of typical round numbers are: 500 million rounds to 5B rounds for regular simulation and hand analysis and

$P_1/P_2=5M/50M$  hands for index generation. There is always a trade-off between accuracy and simulation time. To simulate for example 10B rounds, set the number of rounds to 1B and the *Scale factor* to 10 (on the main window).

Using sampling based index generation the number of rounds should never be less than 5B due to the fact that the results are sampled during normal play.

The templates provided under the *test\_cases* folder are deliberately set up with a small number of rounds to simulate (typically in the range of 10M-50M rounds) for demo purpose to get them to complete in reasonable time on most systems. For any real analysis, these numbers should be increased up to at least 200M or more depending on type of analysis and objective.

In addition to looking at the standard error reported in the simulation results, a practical test that can be conducted to determine if a certain number of rounds is sufficient is to run an analysis or simulation twice with the same number of rounds but with different random generator seeds. Next, by looking at the difference in the results between the two runs, for example in the EV or SCORE metrics, a rough estimate of how much the results may vary can be obtained. This variation gives an intuitive first-order indication of the accuracy in the simulation results.

### 3.1.8 Examples to get started

A summary of all examples and pre-configured templates that come with BJSStrike can be found in Appendix II.

As an example of a typical simulation task, the set-up for a double deck simulation using the Hi-Lo count system and the Illustrious 18 playing indices with a bet spread of 1-6 units is provided in directory:

```
<Installation directory>/test_cases/GUI_Appl/DD1_6_I18
```

Load this session into the simulator using the *Load setting* command and study the options for this simulation in the main window and in the simulation window. The simulation is started by clicking the button *Start simulation* at the bottom of the simulation window.

### 3.1.9 Count system specification

Notice, to enter point values for a customized counting system click on a card tag in the list box and enter the desired value (integer) and press return. In bet variation mode, to enter a bet size, enter the count range and bet size in the three fields and press the ‘add’ button. A dash (-) should be used to indicate a lower or upper bound that corresponds to the minimum or maximum count.

### 3.1.10 Index generation and custom strategy files

Another example that comes with the simulator is an index generation set-up stored in:

```
<Installation directory>/test_cases/GUI_Appl/IndGen_HILO
```

Loading this file and running index generation will generate the most commonly used indices for the Hi-Lo count system (Figure 6). This example provides an easy way of getting started using the index generation. Notice that index generation can be launched for specific 2 or multiple card hands (composition dependent indices) or for total player sums vs. any dealer up card. If total sum is specified, the program will internally generate indices for all 2 card hands that make up the specified player sum and a weighted index based on relative frequency of the individual hands will be reported. Thus, using total sum indices results in significantly longer run time.

In addition, there is new index generation procedure provided in BJSStrike referred to as *Sampling Based Index Generation* (use *Index sampling mode* from the GUI), in which the results of a target hand and the two playing decisions vs. count are sampled during normal play (Section 11.4). Only one hand at a time can be run using this option. The only options from the index generation window that are used in this mode is the count search range and step size in *Force step*.

Customized indices for a playing strategy are specified in the *Custom Strategy* form or in playing *strategy files*, which can specify any composition depended or composition independent indices. An example of a full customized playing strategy file that uses Hi-Lo can be found in:

```
<Installation directory>/test_cases/GUI_Appl/index1_hilo.strat
```

An example of indices specified in the Custom Strategy form can be found in:

```
<Installation directory>/test_cases/GUI_Appl/ENHC_ES10_6D_ILL18.rc
```

To select an existing playing strategy file to be used in a simulation, after checking the ‘*Customized strategy file*’ button on the simulation window, use the ‘*Select file*’ option to select the file. If a file needs to be edited and saved use the ‘*Edit file*’ option to open the file. This is a regular text file so any text editor can be used in addition to the primitive text editor that shows up under ‘*Edit file*’. See also Section 10.3 for details about syntax and what playing strategy constructs are available.

Sections 14.1 and 14.2 describe comprehensive examples on how to generate optimal playing and betting strategies for a card counting system and evaluation of win rate and SCORE for comparison between different systems and strategies.

Strategy file names typically end with the extension *.strat*. Appendix II provides a short summary of all templates and examples that come with BJStrike in the folder:

<Installation directory>/test\_cases/GUI\_Appl/

Notice that the numbers of rounds in these examples may need to be increased for any real analysis since they are deliberately set to small numbers (in the range of 10M rounds only) for demo purpose.

An example of Basic strategy generation can be obtained by loading file:

<Installation directory>/test\_cases/GUI\_Appl/BasicStrat\_A.rc (or BasicStrat\_B.rc for part 2)

To run this example that generates composition dependent optimal strategy for a 6D game at top of deck, open the 'Hand Analysis' window and press 'Start hand simulation'. After completion, the strategy table can be found by scrolling down to the bottom of the results window (View Results). For any real analysis the number of rounds should be adjusted to at least 50M-500M rounds. To generate *counter basic strategy* (i.e. a strategy that considers a betting ramp and penetration in determining the EV maximizing playing decision) for any playing decision the sampling based index generation method can be used (see Section 11.4.2).

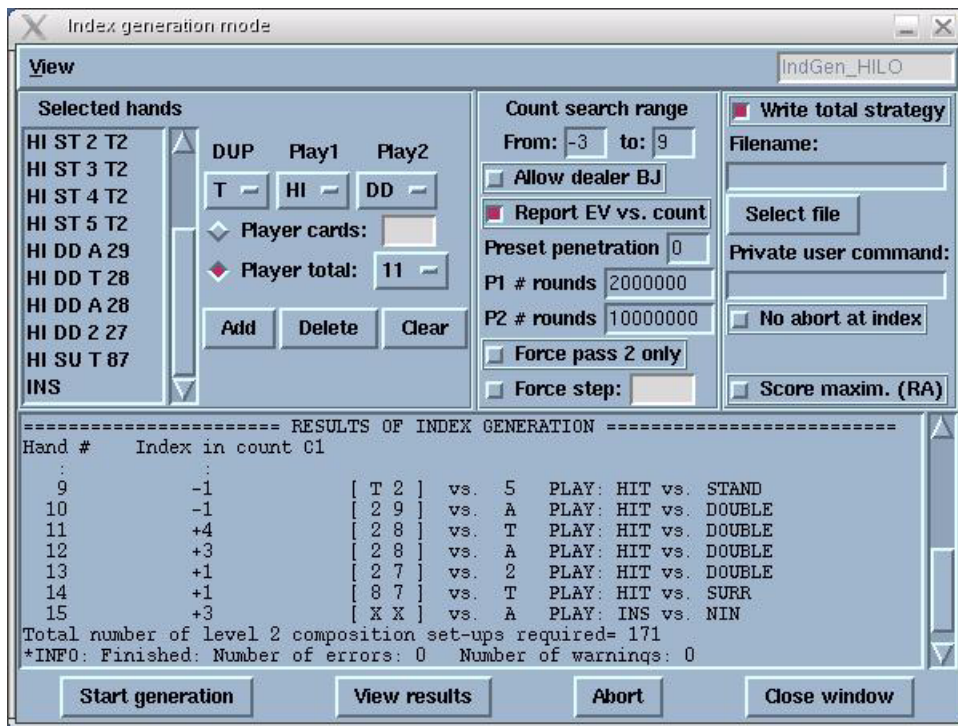


Figure 6 Index generation window using regular index generation

### 3.1.11 Game variations

Several variations of blackjack are supported in BJSStrike. In particular, Spanish 21 is fully supported in simulation and index generation. From the main window selecting “*Game variations -> Spanish 21*” from the main window brings up the Spanish 21 option window (See Figure 31 in 13.1). Before using Spanish 21 it is recommended to read Section 13.1 in detail.

### 3.1.12 Details of GUI operation

During normal operation, when simulation or index generation is started the GUI first generates an ordinary command file based on the current setting. Next, the GUI invokes `bjstrike` as in normal command line invocation. The text generated on standard output is directed to the text frame of the application window. To view the result report after a completed simulation, press the “*View results*” button and the results of the latest run will show up in a separate text window. The command file generated for any GUI simulation task is locally written into file:

```
<workspace>/tcl_arg_f.rc
```

where `<workspace>` is the directory specified in the environment setting as workspace. This file can be inspected to learn about what command line options are generated for a certain GUI setting.

A local script, which invokes the simulator, is generated on the fly under the file name:

```
<workspace>/run_bjsim
```

This script can be used as an example of how to run `bjstrike` in command line mode for batch mode based applications.

In case any of the text commands described in Section 5 is desired to be used in addition to the normal options provided by the GUI, the entry field named *Private user command*<sup>1</sup> on the index generation window can be used. Any text string written there will be passed directly to the simulation engine and interpreted as a regular command. This means that any of the commands described in Section 5 can be directly entered from the GUI. In addition, special customized commands requested by users which are not officially documented in Section 5 are also entered here since these options may not have dedicated entries in any window. The *Private user command* entry is good for only one command (or a reference to a command include file that can contain any number of commands using ‘++ *<File name>*’).

In case several text commands are desired, it is recommended to use a strategy file. Any set of options and commands describe throughout this manual can be entered directly in a strategy file if desired using the format:

```
//++ COMMAND
```

Example:

```
//++ DEF_COUNT C5 HILO INIT=0
```

---

1. Notice that the command ‘++ *<File name>*’ can also be entered in this field. This command can be used to specify an argument command file by name that will be read into the simulator to allow a sequence of custom commands to be specified.

The strategy file now contains one or more commands possibly in combination with a custom playing strategy. The strategy file is easily referenced from the simulation window by checking *Custom strategy file* and *Select file* (See Section 10.3.4).

### 3.1.13 Multi-thread simulation

Multi-thread simulation is useful for speeding up simulation in multi-core or multi-thread systems. Multi-thread simulation mode is invoked from the *Environment* menu of the main window. 2-32 threads can be specified. Default 0 means single thread simulation. Since the level of parallelism that can be extracted from a Blackjack simulation task is very high, an almost linear speed-up of simulation performance with number of processor threads can be achieved. Multi-thread simulation is currently support for regular simulation and sampling based index generation for the following operating systems: Linux, Mac OS X and Solaris.

It is also possible to run multiple sessions of BJSStrike simultaneously to gain speed-up in simulation throughput on multi-thread systems.

### 3.1.14 Fine tuning the GUI's look-and-feel

The color of the windows in BJSStrike can be customized as the user wishes the following way. Open file *bjst.tcl* in the main installation directory using a text editor. Then, search for a line that contains the word *tk\_setPalette*. Change the argument of this function to whatever color name is desired. For example, edit the line to look like “*tk\_setPalette red*” for forcing all BJSStrike windows to the background color red. A list of all valid color names can be found on-line under the BJSStrike support section. Updates to the GUI are available on the support web page periodically from the link provided in the Software cover letter.

### 3.1.15 Compatibility with older versions of BJSStrike

Setting files saved in earlier versions of BJSStrike and strategy files used are fully compatible with newer versions, meaning that any setting file you have created in an earlier version should work as is in BJSStrike 4.0. A note about the older BJSStrike versions. BJSStrike version 3.2 and older do not work with the latest version of the graphical toolkit (tcl/tk 8.6). Any older BJSStrike requires tk version 8.5 or older. On the other hand, BJSStrike version 4.0 will work with any toolkit version later than 8.3.4.

## 3.2 Command line invocation

From the command line, the simulator can be invoked through the command:

```
bjstrike [<Strategy file>] { Option list ... }
```

See Appendix III for information about how to set the shell environment variables that are needed to run BJSStrike in terminal mode.

**<Strategy file>**      Optional input file  
                           This file can be used to specify a user customized playing strategy including playing indices for count systems.

### 3.2.1 Command line options

- argf <file name>** Specifies the argument file to be used. The argument file contains user commands, configurations, rule setting and task definitions. If not specified the default argument file, *vdsim\_def.rc* located in the installation bin directory will be read. The default argument file specifies a setting for a single deck H17 simulation.
- lic** Prints license and node lock information about current simulator version and then aborts.
- h** Prints information about invocation arguments
- bindir <Directory>** Optional: Overrides the environment variable DFTSIMULAB\_BINDIR to specify the bjsstrike executable directory, required for Windows.
- lf <License file>** Optional: Overrides the environment variable VDLSIM\_LM\_PATH to specify the bjsstrike license file, required for Windows.
- homedir <Directory>** Specifies user homer directory for Windows. Required for Windows. Must point to the directory that contains the reg. file. Must be the same as returned by `tcl env. HOME` or `env(USERPROFILE)`  
Example: `-homedir C:\Users\`
- defdir** Optional: Specifies a directory to be used as the results directory
- regcode <PIN>** Registration of simulator through command line (Unix/Linux)
- regcode <PIN> <User home directory>** Registration of simulator (Windows)

Example of command line invocation:

```
/usr/local/lic/bjsstrike -argf SD_H17_ILL18.rc
```

To run in batch mode (command line mode) under Windows there are examples of command files to run under:

```
{installation directory}\test_cases\TC_SD_H17_SIM
```

Depending on version of Windows one of the batch files: `dos_sd_flat*.bat` should work. It is assumed that the software has been registered with a PIN prior to running in batch mode.

For example for Windows XP, the command file to run is:

```
dos_sd_flat2.bat
```

### 3.2.2 Simulation results

Most status information including error and warning messages are written both on the screen (standard out) and in a log file. The name of the log file is fixed to *VDSim.log*. The log file typically contains more detailed messages and most reports generated by the simulator are written in the log file only. As the log file gets overwritten for each run, it's up to the user to copy a log file that contains results of a simulation he wants to save. It's also advisable to copy the argument file that was used to generate the results together with the result files. When using the GUI, these files are automatically saved when *File -> Save setting as* is performed

## 4 Practice mode

The practice mode is useful for a beginner to learn the game of blackjack as well as for the expert to practice advantageous play techniques including card counting.

This mode allows you to play the game of blackjack against a computer dealer under many different conditions. A variety of training options are available for practicing counting, playing decision and betting. As the play and practice module has been added on top of the BJStrike simulation engine, most of the options available for simulation are also available in playing mode including all the strategy and betting options as well as deck composition options and most of the rule variations.

Play one or multiple (1-5)<sup>1</sup> hands at a blackjack table or just watch a game being played by all computer players for practicing back-counting. Auto play mode can be used for realistic card counting drills. There are 20 parameters to customize the speed of a game or simply use any of the four standard speed option templates.

### 4.1 Practice mode option window

Selecting the “PRACTICE” button from the main window causes the practice option window to show up, see Figure 7. Most options related to practice and play can be set up here. In addition, most of the regular options available for simulation in the main window (Figure 1 of Section 3.1) and simulation window (Figure 2) that are related to rules, deck composition, counting systems, betting and playing strategy will have the same meaning in the practice mode. A game is started by pressing the “Start game” button. Notice that a game in progress has to be aborted if any options need to be changed. Bjstrike will return to the main window if a game is aborted. From there, the practice option window can be re-launched to start a new game.

#### 4.1.1 Table spot and number of players

In the first column of options in Figure 7, the number of players can be specified. In addition, there are check boxes for each table spot. Checking any or several spots means that these spots will be played by the player, you. If none of the table spots are selected the game will run in auto play mode with all spots played by the computer. In this mode, you can watch a game in progress for practising counting and the program will ask you for count at certain round number (if drill option is chosen). All computer players will play and bet according to current active strategies, which are set up in the same manner as in simulation. The only way to abort a play in progress in auto mode is by pressing the abort button in the playing table window.

#### 4.1.2 Computer optimal play

Checking this option, results in the simulator calculating the optimal playing decision for each decision based on current deck state. This is the decision that will be used as the computer correct play regardless of what strategy is chosen in the simulation window. This is useful in evaluating perfect play in simulation and practice as well as in evaluating count system playing

---

1. Depending on the screen size and system platform up to 6 hands can be used in play mode.

efficiency. Any deviation from optimal play by the strategy chosen is reported and can be observed in the log file on a shoe-by-shoe basis.

There are restrictions as to what hand types are evaluated. For example, insurance and any hand involving split or resplit decision is not evaluated under computer perfect play. For more details on computer perfect play options see Section 10.6.

#### **4.1.3 Dealer card pick up order**

The check box “R to L” at the bottom of the first column represents the default order of card collection by the dealer. This dealer procedure means that the cards are turned over (for face down games) and picked up by the dealer starting from his rightmost player spot (third base) and ending at his left player spot (first base). The other option “L to R” selects the opposite order of card collection.

#### **4.1.4 Action on playing decision errors and betting errors**

What BJStrike should do in case it encounters a playing decision from the user, that is not the optimal play according to the active strategy, is specified in the second column of options. In *auto correct* mode, a warning message will show up in a small window with the correct play indicated. The user must then press the OK button and BJStrike will change the play to the correct play according to the strategy and rules. In *Inform* mode, BJStrike will only write a short warning message in the dealer message text line and the user’s playing decision will be used.

Similarly for betting errors, *Auto correct* mode means that a warning message will pop up with the correct bet indicated. Next, upon pressing Ok, the correct bet amount will be used.

Notice, that BJStrike is set up to use \$25 as the default bet in flat bet mode in practice and play mode, which is different from the default flat bet of 1 unit used in simulation. This, of course, can be changed in the betting option windows if desired.

Statistics about number of playing decision and betting errors can be inspected by pressing the Info button from the blackjack table window (see D of Figure 8) or after a session has been finished from the menu “View->View practice results” of the BJStrike main window.

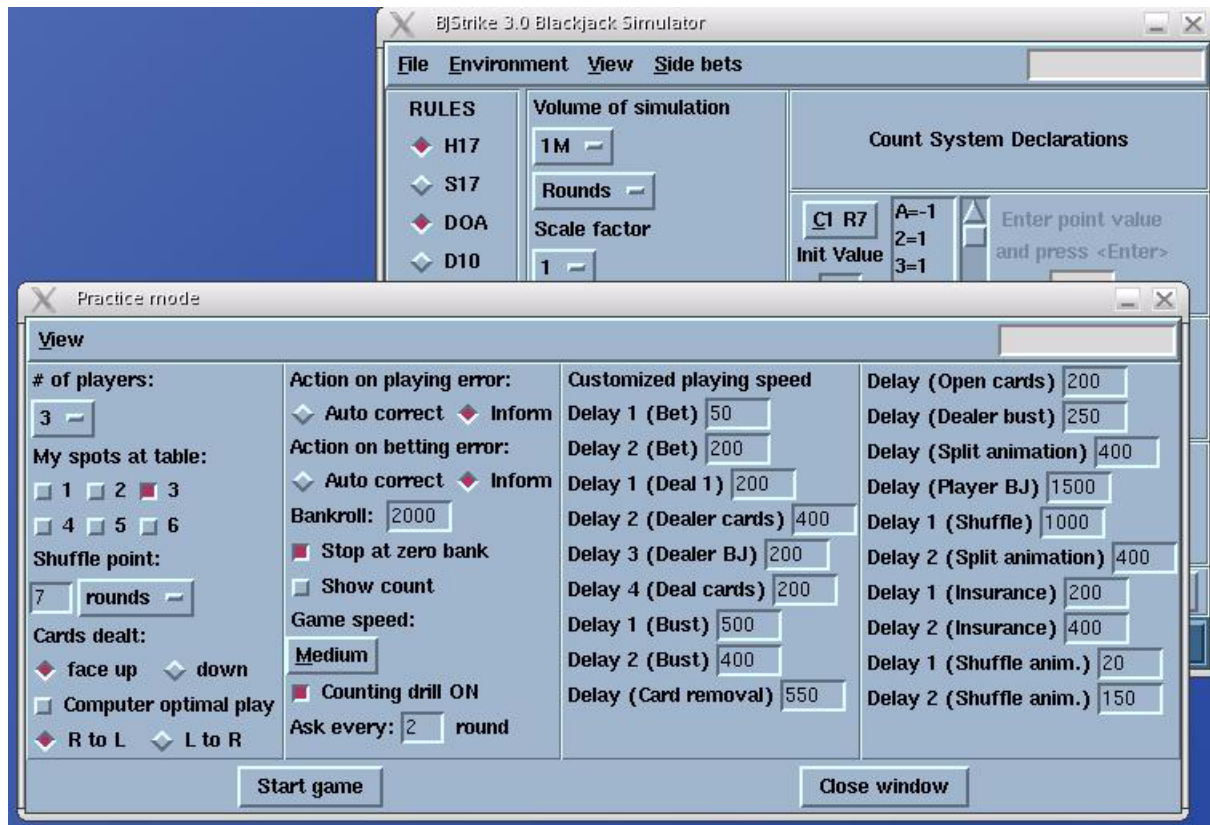


Figure 7 Practice option window

#### 4.1.5 Bankroll options and Show count option

The bankroll specified applies to each hand. The bank of each player is visible at each player spot (C of Figure 8). The default initial bankroll is \$2,000. If the “Stop at zero bank” option is checked the game will finish as soon as a human player spot reaches a value below zero. If this option is not checked negative bankrolls are allowed during play and the game will continue as normal even at zero bank.

Checking the *Show count* option will cause the current True Count (TC) and Running Count (RC) to be displayed for a short time at the beginning of each round. Notice that TC and RC can always be viewed in the Info window that shows up when pressing the “Info” button on the practice window. This window can also be launched when the play is in playing decision mode.

The count system used is count series C1, that is specified in the main window. If C1 is inactive no count related information is displayed.

#### 4.1.6 Game speed options

There are four standard speeds (ultra slow, slow, medium, fast). In addition, there are 20 different delay parameters that control various animation procedures during play that can be individ-

ually tuned. Milliseconds is used as time unit for all delay parameters shown in the third and fourth columns of the practice option window.

#### 4.1.7 Counting drill options

Checking the *Counting drill on* option will result in the simulator asking the user for the true count (or RC) at certain round number specified in the entry *Ask every ? round*. A window will pop up when the round number is satisfied as shown in Figure 10(b). The user then enter the count and press the Done button. Statistics about number of questions answered and number of errors are displayed in the same window. Entering the value one in the “*Ask every ? round*” entry will result in the count being checked at the beginning of each round except for the very first round after a shuffle.

Notice that a count system has to be declared in the simulator main window for count series C1 before starting the game in order for the counting drill option to work.

Back-counting a game in progress can be practised by unchecking all playing spots (i.e. all players will be computer player) and setting up a speed that is appropriate for the game to emulate. Next, while the play is in progress, the count will be asked for at the specified round number interval.

## 4.2 Playing window

Pressing the start game button from the practice mode option window will launch a playing table window as shown in Figure 8 (for three players and one user spot in this example).

During normal play, there are three user input modes in which different user inputs are possible: (1) Betting input; (2) Insurance decision; (3) Playing decision.

There are dedicated buttons to signal most of the actions in play. In addition, there are shortcut key bindings and mouse bindings that are active for the various user input modes, which are listed in Figure 9. This shortcut list will also show up any time the *Help* button is pressed. To leave the game and return to the BJSStrike main window the *Abort* button should be pressed. The *Info* button will provide statistics about the practice session in progress and suggest playing decision as well as provide information on penetration and TC. Notice that, the *Info* button can only be chosen in playing decision mode (3) in which the button is high-lighted. An example of the *Info* window is shown in Figure 10(a).

### 4.2.1 Betting input actions

For the very first time a betting amount needs to be selected using the betting amount buttons to add and subtract chips from the player bank to the betting circle. In addition to these buttons, pressing the right mouse button or *<Enter>* is a short cut for betting the same amounts as in the previous round for all user hands.

### 4.2.2 Insurance decision

In case of an insurance situation, each player hand must answer either yes (y) or no (n).



- A) System message area for betting/strategy errors and warning messages, TC and dealer messages
- B) Amount currently in the betting circle
- C) Bankroll per player spot
- D) Info button, will display practice statistics, TC, penetration and optimal playing strategy (Can be invoked in playing decision mode only)
- E) Displays 2 numbers: # of shoes played / # of rounds played

**Figure 8** Example of playing window for three players of which one is the user hand

### 4.2.3 Playing decision actions

In playing decision mode, in addition to the buttons for the various playing decisions, the shortcuts as shown in Figure 9 can be used to signal a certain play. For example, double clicking on the left mouse button signals a hit and a single click on the right mouse button signals stand.

Notice that the **Info** button that launches the information window can only be invoked in the playing decision mode.

Some things to keep in mind in playing decision mode:

- The player must always signal a decision even if the total sum is 21. Rules RSA, DAS and SU can always be chosen even if these rules are disabled in the main simulation menu. On the other hand, choosing any of these plays while disabled will be considered a playing decision error and handled according to what's specified for the action on error options in the practice option window.
- Player is allowed to hit on splitted aces and must physically signal a stand to follow the rule of only one card hit on split aces.
- The cards of busted split hands as well as the chips are collected **after** all splitted hands for a player have been completed and not immediately at the time of the bust, which is commonly done in real play.
- The following rule variations are not supported at all in practice mode and are therefore ignored: ENHC, BJ payoff other than 3:2, resplits to more than 4 times, bonus hands, side bets and no more than 9 hit cards per hand is allowed.

```
BJStrike practice mode command shortcuts and key bindings
=====
```

```
1) Betting mode
<Button-3>, <Button-2> or <Enter/Return> ==>
    Bet same amount as last round on all user spots

2) Insurance decision mode
Accepting insurance can be signaled by: Keys: <i/I>, <y/Y>
Rejecting insurance can be signaled by:
    <Button-2>, <Button-3>, <Enter/Return> or <n/N>

3) Playing decision mode
<Double-click-Button-1> or <h>          ==> Hit
<Button-2>, <Button-3>, <Return> or <s> ==> Stand
<d> ==> Double Down
<p> ==> Pair split
<u> ==> Surrender
```

```
Notice: Button-3 events can be generated by pressing <ctrl-Button-1>
on a single button mouse such as Mac
```

```
-----
```

```
The <Info> button can be pressed in Playing decision mode (3) only
and will provide information on expected play, count and penetration.
```

**Figure 9** Key and mouse bindings in practice mode

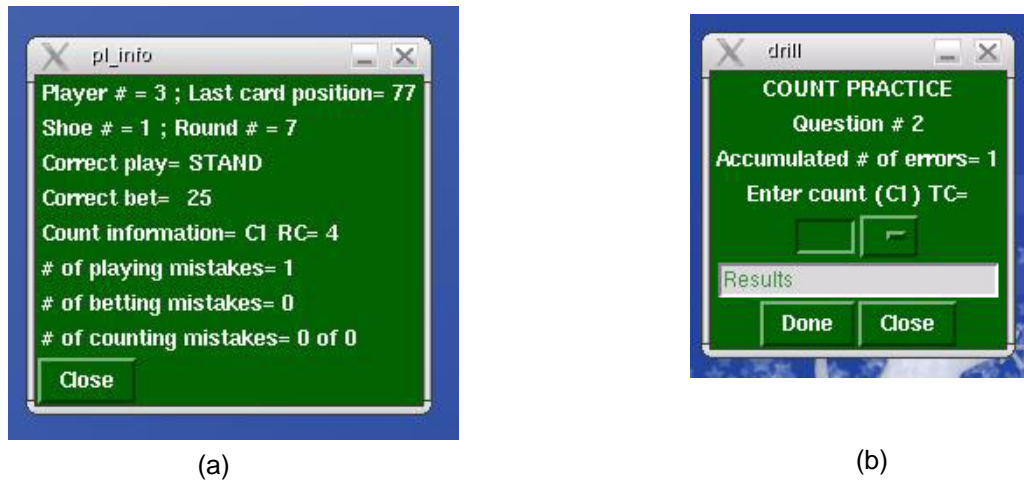


Figure 10 (a) Playing information window and (b) Count drill window

### 4.3 Things to keep in mind in practice mode

Most playing options with the exception of the speed parameters are saved using the regular “Save setting” option in the File menu of the main window. Each round played is logged and saved in the regular result file (VDSim.log) and can be inspected under the “View”->*View sim. results* menu of the main window at any time. This is useful if the user wants to go back and look for how he or the computer players played on earlier hands. As in normal simulation, the entire results file is saved using the “Save setting” option in addition to the settings. For each hand played, all cards drawn as well as betting and playing decision can be observed in this result file. This means that results and play action are written to the log file continuously as the game is played.

Notice that the *Info* button that launches the information window shown in Figure 10(a) only works in the playing decision mode. The *Help* button, on the other hand, can be used any time and will bring up the list of all key and mouse button bindings as shown in Figure 9.

In a game in progress, to return to the main window, the *Abort* button must be pressed.

Notice that after a practice session is completed the results of the practice session in terms of number of playing betting and counting errors can be inspected under “View”->*View practice results* menu. This option will bring up the window shown in Figure 10a. Notice that all playing results and practice results can be saved for later analysis using the “File”->*Save setting as* menu of the main window.

Only *normal* Blackjack rules are supported in practice mode, i.e. special rule variations such as SP21 redoubling, rescue surrender, dealer wins tie, multiple card DD, bonus hands, HSA and ENHC cannot be used.

## 4.4 Templates provided for practice and play mode

BJStrike comes with several pre-defined settings for different practicing sessions. These settings can be found under: *test\_cases/GUI\_Appl/* in the distribution folder. Load one of these files using the *File->Load Setting* menu from the main window before invoking the PRACTICE mode.

- **Practice\_BS\_6D.rc**                    Setting for practicing basic strategy for a S17, 6D game. Will signal and correct any playing error.
- **Practice\_hilo.rc**                    Setting for practicing the Hi-Lo count system in a DD game. TC conversion accuracy=0.5
- **Flash card mode (auto play)**    Load the **Practice\_hilo.rc** setting and uncheck box 3 under “*My spots at table*” option of the Practice option window. Select an appropriate playing speed and the game will play without user interaction with the drill window showing up and asking for true count at certain round numbers.
- **Practice\_Perf\_SD.rc**                Practice setting for computer perfect play as the strategy used. This example is for Single deck flat bet.