1993 AI Lab Computer Poker Night

Mark Torrance
torrance@ai.mit.edu

January 14, 1993

1 Introduction

This manual describes the rules, software environment and support programs for Computer Poker Night, an event in the 1993 AI Lab Olympics. Participants will write programs to play five card stud poker against each other and against people. These programs can be tested during January by connecting to a "Casino Server" which will always be running, populated with dupe computer players eager to be taken to the cleaners. Participants may wish to coordinate practice times for their programs to play against each other on the casino server. People may also play against the programs or each other using a human interface, provided. *The final contest will be held at 7pm on Wednesday, January 27 in the 7ai playroom.*

The game we will be playing is five card stud poker, as described in *The New Complete Hoyle Revised* by Morehead, Frey and Mott-Smith. The play of the game is described in Section 6 at the end of this document.

Nomenclature

In this document, cards are named by rank followed by suit: **A♠** is the ace of spades and **2♥** is the two of hearts. In the computer programs, cards are named by suit followed by rank: `s1` is the ace of spades, `h2` is the two of hearts, and `d12` is the queen of diamonds.

2 Casino Server

A program, the Casino, has been written and will be running throughout January. The Casino maintains a number of *tables*; at each table any number of players from two to eight may play stud poker. The Casino will maintain as many tables as are required, dynamically creating new tables or destroying empty tables in order to keep exactly one empty table available at all times.

Users can connect to the casino on `cookie-monster.ai.mit.edu` on port `1969` (not yet).

3 Human Player

People may use a human player interface to enter the casino. This program is called `poker`, and is available on the AI Lab machines as `/usr/local/bin/poker` (not yet).

4 Computer Players

Participants will want to write their own computer players for the game. Some sample computer players are available, written in TCL and C.

Player programs communicate with the casino over network TCP connections. The protocol is based on the messages described below. The sections about the available programming language interfaces, which follow, describe the syntax of these messages as it is implemented in those languages.

4.1 Messages to Casino

Messages from players to the casino by convention begin with a lowercase letter. These messages include:

tables ***Function***
Ask the casino what tables are available and who is playing at them. Returns a list including the word *TABLES*, followed by an entry for each existing table which has keys indicating the amount of money in the pot at that table, the players at that table (and their cash amounts), and the waiters at that table (and their cash amounts). For example, *TABLES* {{*pot 10*} {*players* {{*mike 10025*} {*mark 9965*}}}} {{*pot 0*}}. For purposes of the *joinTable* and *watchTable* commands, tables are numbered in the order they appear in this list. In the example, *mike* and *mark* are playing at table 1 and table 2 is empty.

watchTable *n* ***Function***
Watch the game at table n. May return *ErrAtTable* if you need to disconnect from your current table first, *ErrNoTable* if that table does not exist, or *Watch* to tell you that you are watching table n. Note that the number of a table may change over time as empty tables are created and destroyed; it is best to send a *tables* message to verify the number of the table you choose to watch just before you send the *watchTable* message.

signon *name password* ***Function***
Register yourself with the Casino and receive chips, enabling you to join a table to play poker. Returns the number of chips you have been allocated. Before the night of the contest, any password will work. During the contest, only certain one-time-only passwords will work; these will be distributed to participants by the poker subcaptain from their team.

joinTable *n* ***Function***
Join the queue of waiters at table n. May return *ErrSignon* if you need to signon first, *ErrAtTable* if you need to disconnect from your current table first, *ErrNoTable* if that table does not exist, or *Join* to tell you that you are waiting at table n. Note that the number of a table may change over time as empty tables are created and destroyed; it is best to send a *tables* message to verify the number of the table you choose to play at just before you send the *joinTable* message. Table 1, the first table, will always exist and will be the table you should play at unless there are many other players.

exitTable ***Function***
Join the list of leavers at your current table, saying that you wish to leave the table as soon

as possible. May result in *ErrSignon* if you never signed on, *ErrNotAtTable* if you are not connected to any table, or *Leaving* if you have been added to the list of leavers. You will leave the table as soon as you are not participating in a game, and at that time everyone will receive a *Leave* message. If you are playing and you want to leave immediately, you should wait for your turn, send a *fold* message and then an *exitTable* message.

fold ***Function***
When it is your turn, send this message to say you want to fold. The casino will return either *ErrNotYourTurn*, or nothing if successful.

call ***Function***
When it is your turn, send this message to say you want to call. The casino will return either *ErrNotYourTurn*, or nothing if successful.

raise *n* ***Function***
When it is your turn, send this message to say you want to raise. n should be the *additional* amount you want to raise by; your total bet for this turn will be the amount needed to call, plus n. The casino will return either *ErrNotYourTurn*, or nothing if successful. If $n \leq 0$, your move will be *call*. If $n \geq maxbet$, your move will be *raise maxbet*.

4.2 Messages from Casino

Messages from the casino to the players are by convention capitalized. These messages include:

Ante n **Every player antes** n ***Message***
Each player has contributed n to the pot. Indicates the beginning of a new hand.

YourTurn j m n ***Message***
It is your turn to bet. You should bet j to call, your maximum raise amount is m, and you currently have n chips. You should send *fold*, *call*, or *raise* shortly after you receive a *YourTurn* message.

Cards {**name1 up** {**c1** ...}} {**name2 up** {**d1** ...} **down** {**d2**}} ***Message***
This message indicates that you can see the two upcards of player *name1*, and the two upcards and one downcard of player *name2*, which is you (since you can see downcards).

FinalCards {**name1 up** {**c1** ...} **down** {**c2**}} {**name2 up** {**d1** ...} **down** {**d2**}} ***Message***
This message has the same format, but will include the upcards and downcards of all players still in the game; it is sent just before the winner is determined. Note that if all players but one drop, no *FinalCards* message is sent because the last player is not required to show his downcards.

Folds name has folded ***Message***
Player *name* has folded.

Calls name j ***Message***

Player *name* has called by putting in j chips.

Raises name j r ***Message***

Player *name* has raised by putting in j chips to call and r more to raise.

Winner name wins p **with hand** ***Message***

Player *name* has won the pot of p chips, with a hand described in *hand*. Possible descriptions include *High Card*, *A Pair*, *Two Pair*, *Three of a Kind*, *Straight*, *Flush*, *Full House*, *Four of a Kind*, *Straight Flush*, *Royal Flush*, and *Others Dropping Out*.

LowCash Not enough cash for ante ***Message***

At the beginning of a hand you did not have enough cash left to ante and you have been demoted to a watcher at that table. You will also get a *Watch* message about yourself.

LowCash Not enough cash to call ***Message***

In the middle of a hand you did not have enough cash left to call and you have implicitly folded and been demoted to a watcher at that table. You will also get a *Watch* message about yourself.

Join name is waiting at table i ***Message***

There is a new player in the wait queue at your table.

Leave name has left table i ***Message***

Player *name* has left your table.

Play name is now a player ***Message***

Player *name* has been promoted from the wait queue to become a player at your table.

Watch name is now a watcher ***Message***

Player *name* has become a watcher at your table.

4.3 TCL interface

TCL, which stands for Tool Command Language, is an interpreted string-based language which makes it easy to write toolkit applications that work together, including TCP connections and interactive Motif-style graphical user interfaces. The Casino server, the human interface program `poker`, and a sample computer player `tclplayer` are written in this language.

TCL programs are run by first running a TCL interpreter, or shell, and then loading the TCL program into the running shell. TCL programs that use graphics, such as the human player `poker`, are run under the shell `wishx-tcp`. TCL programs that don't use graphics may be run under the shell `tclxtcp`. Note that programs which use graphics are not secure; it is possible for anyone who can connect to the X server on which the program `wishx-tcp` is running to issue arbitrary commands to be interpreted by the running graphical TCL program. In particular, malicious users could determine your cards if they can figure out what you named the variable in which you store them. Non-graphical TCL programs are as secure as any other UNIX program; it is advisable to have a non-graphical version of your program for the contest. The casino server is a non-graphical TCL program that runs under `tclxtcp`.

To write a computer player in TCL, it is best to start from the sample player defined in `/com/olympics/poker/tclplayer` (not yet available). In TCL, each message that the player sends to the casino is implemented by doing a remote procedure call on the casino server with the text of the message as the procedure and arguments. Thus, once you have established a connection to the casino (see the `enter_casino` procedure in the sample player), you would send the signon message with `[tell_casino "signon` *name password*`"]`. There are examples of all the message types in the sample player.

4.4 C interface

This section needs to be written

4.5 Lisp interface

This section needs to be written

4.6 Scheme interface

This section needs to be written

5 Scoring

On the night of the contest, a scoring program will be run which can display running graphs of the number of chips each player in the game has as a function of time. This program will display the results on a public display in the 7ai playroom so all can see how the programs are doing. This program will be implemented as another player connected to the casino which goes around watching tables.

Awarding of Olympic points to teams will be determined and described here.

6 Poker Laws for AI Lab Computer Poker

6.1 Players

Any number of players from two to eight may play in a single Poker game.

6.2 Cards

The pack of 52 is used, with the cards in each suit ranking **A** (high), **K**, **Q**, **J**, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **2**. There is no rank of suits.

6.3 Objects of the Game

Each player endeavors to hold a better *poker hand* than any other player in the game. A poker hand consists of exactly five cards.

6.3.1 Rank of Poker Hands

The following list states the combinations which make up valuable poker hands, and their rank. The combinations and rankings given are standard for virtually all Poker games.

Straight flush ranks highest. It consists of five cards in suit and sequence, with the ace ranking either high or low: **A♦**, **K♦**, **Q♦**, **J♦**, **10♦** (the highest possible hand, also called a *royal flush*), or **5♦**, **4♦**, **3♦**, **2♦**, **A♦**, or any sequence in between, as **10♠**, **9♠**, **8♠**, **7♠**, **6♠** or **8♣**, **7♣**, **6♣**, **5♣**, **4♣**, is a straight flush.

Four of a kind is next highest. It consists of the four cards of any one rank together with any fifth card; for example, **7♥**, **7♦**, **7♣**, **7♠**, **Q♦** constitute four sevens.

A *full house* ranks next; it consists of any three of one kind and pair of another kind, as **6♠**, **6♥**, **6♦**, **A♣**, **A♠**. It is referred to by the three of a kind it contains; the example shown would be "sixes full."

A *flush*, ranking next, consists of any five cards of the same suit, but not in sequence, as **J♦**, **9♦**, **8♦**, **6♦**, **3♦**, referred to as a "jack-high flush."

A *straight* consists of any five cards of two or more suits in sequence of rank, with the ace ranking either high in the sequence **A-K-Q-J-10** or low in the sequence **5-4-3-2-A**. It ranks next under a flush.

Three of a kind are any three cards of the same rank plus two other cards which do not constitute a pair and do not include the fourth card of the same rank; **9♠**, **9♦**, **9♥**, **K♣**, **3♦** would be referred to as "three nines."

Two pairs, which rank next under three of a kind, constitute two cards of one rank, two cards of another rank, and any fifth card which is neither of thoseee ranks; it is referred to by the higher of the two pairs. Thus **Q♠**, **Q♣**, **8♠**, **8♣**, **4♠** would be "queens up."

Any two cards of the same rank, together with three other cards which do not combine with the other two to form any of the higher-ranking hands listed above, are a *pair*; **K♠**, **K♣**, **7♥**, **6♥**, **4♥** are a pair of kings.

No pair—the lowest ranking hand—losing to any hand containing a pair or any better combination, consists of any five cards not meeting the specifications above.

6.3.2 Object of Betting

The players in the game bet with one another as to which has the best poker hand. Each deal is a separate game, in that its result does not affect any preceding or subsequent deal. All the bets are placed together and form a *pot*. A player who does not wishhh to bet that he has the best hand may *drop* (or *fold*), thus relinquishing any chance to win the pot.

The ultimate object in Poker is therefore to win the pot, whether by actually holding the best hand or by inducing other players to drop and leave the pot to be taken, uncontested, by a single player still willing to bet.

6.4 Sequence of play

6.4.1 Starting to play a game

When a player connects to the casino, he or she will need to *signon* to receive chips. At signon time, the player must provide a password. During practice sessions throughout January, players can receive chips with any password. At the final contest, each AI Olympic team will receive twenty passwords, each good for one and only one login. The poker subcaptain for each team will determine how these passwords will be distributed to team participants.

Between two and eight players may play at each *table* in the casino. Players can ask what tables are available with the command *tables*. Players can watch the proceedings at a table with the *watchTable* command. Players can join a table with the *joinTable* command. If a player joins a table while a game is in progress, he will be put in a queue of *waiters*. Once at least two players are waiting at a table and no game is already in progress, a new game will be started. Each time a game ends, waiters will be promoted from the head of the queue to become players until the table is full (8 players). All players, waiters and watchers at a table are *recipients*; they will receive all messages from the casino about the cards they can see, the sequence of play, the actions of the players, and the winner of the game.

6.4.2 Rotation

Players are seated at the table in first come first serve order. New players are always added in the order after the newest player already seated at the table. When a player leaves the table, the players who flanked him become neighbors.

The cards as they are dealt and the turn to bet both pass from player to player to the left. Once a player has dropped (folded), the turn skips him and takes up with the next player to his left who has not dropped.

6.4.3 Chips

The pot is formed before the deal and begins by the contribution of an *ante* from each player.

6.4.4 Procedure

When a new game begins, each player receives one face down card and one face up card. All *recipients* at the table are informed of the cards they can see with a *Cards* message. The player with the highest card showing bets first to begin the first *betting interval* (see below in section 6.6). Betting proceeds around the table in cyclic order of arrival in the game. Betting continues until the end of the betting interval or until all players but one have folded. Next a third card is dealt face up to each player that hasn't folded and another betting interval follows. This is followed by a fourth face up card, another round of betting, a fifth face up card and a final round of betting.

At the end of the last betting interval is the *showdown* at which each player who has not previously dropped has his "hole" cards (face down cards) revealed with a *FinalCards* message from the casino. Whichever of these players has the higher ranking poker hand is announced as the winner and given the pot by means of a *Winner* message from the casino. At the end of the game, all players who are waiting to leave the table are allowed to leave, and a new game is started if there remain at least two players.

6.5 The Shuffle and Deal

The casino gives one card to each player in rotation face down; then a second card to each player face up. A player's face-down cards are his "hole cards" and are not shown until the showdown. Each recipient gets a *Cards* message informing them of all the cards they can see.

After these cards have been dealt, the deal is interrupted for a betting interval. There are then three more rounds of cards dealt face-up, one at a time, with a betting interval after each. After the final betting interval, a *FinalCards* message reveals the hole cards and the casino distributes the pot with a *Winner* message.

6.6 Betting

In five card stud poker there are four *betting intervals*. The player whose turn it is to bet receives a *YourTurn* message informing him of how much he would need to put in to call, his maximum raise, and his current amount of cash.

In each betting interval, one player in the game has the privilege or duty of making the first bet. Each player in rotation after him may either

(a) *drop*, by discarding his hand; in this case, he no longer participates in the pot, and cannot win the pot regardless of what may later occur;

(b) *call*, which means that he places in the pot enough chips to make his contribution to the pot *during that betting interval* as great as the contribution of any other player but no greater;

(c) *raise*, which means that he places in the pot enough chips to call, plus one or more additional chips.

To *illustrate this*: There are seven players in the game, designated as A, B, C, D, E, F and G. A is the first bettor. A bets one chip. B calls by putting in one chip. C drops, discarding his hand. D raises, putting in three chips—one chip to call and two to raise. E calls, putting in three chips. F raises, putting in six chips–three to call and three to raise. G drops.

It is now the turn of A again. A calls, putting in five chips; this makes his total contribution to the pot six chips during this betting interval. B drops; he relinquishes his chance to win the pot, and the chip he previously put in the pot remains there. D calls, putting in three chips to bring his total contribution up to six. E raises five chips by putting in eight chips. F raises by putting in ten chips. A drops, thus losing the six chips he had already put in. D calls by putting in ten chips. E calls by putting in five chips, and now the betting interval is ended. D, E and F are still in the pot, and each during this betting interval has put sixteen chips into the pot.

A betting interval ends when every active player has had at least one turn to bet and when the bets have been *equalized*—that is, when every player has either contributed the same amount as every other player has contributed during the betting interval, or has dropped.

6.7 Checking

A player in his turn is permitted to make "a bet of nothing" by saying "Check," provided no previous player has made a bet during that betting interval. The effect of the check is that the player merely wishes to stay in the pot by making a bet so insignificant that it is not worth the trouble of putting that amount into the pot. If the first bettor checks, each active player thereafter may check until any player chooses to bet, after which a player may stay in the pot only by at least calling the previous bet. Raising after you have checked is legal. *To check in AI Lab Poker, your opening play should be* call.

6.8 The Showdown

When the bets have been equalized in the last betting interval, every player who has not previously dropped must expose all his cards face up on the table. The casino does this with a *FinalCards* message. The highest-ranking poker hand wins the pot. The casino does this with a *Winner* message.

If at any stage of the game every player but one has dropped, the latter takes the pot without showing any of his cards.

6.8.1 Breaking Ties

When two players have hands of the same type, the higher-ranking hand is determined as follows:

If each has a straight or straight flush, the one including the highest card wins (**6**, **5**, **4**, **3**, **2** beats **5**, **4**, **3**, **2**, **A**). If each has three of a kind or four of a kind, the one composed of the higher-ranking cards wins.

If each has two pairs, the one with the highest pair wins; **A**, **A**, **3**, **3**, **4** beats **K**, **K**, **Q**, **Q**, **5**. If each has the same higher pair, the hand with the higher of the two lower pairs wins; **A**, **A**, **6**, **6**, **5** beats **A**, **A**, **5**, **5**, **9**. If each has the same two pairs, the one with the higher fifth card wins; **A**, **A**, **6**, **6**, **7** beats **A**, **A**, **6**, **6**, **5**.

If each has one pair, the higher pair wins. If the two pairs are the same, the winner is determined by comparison of the other three cards in each of the two hands, depending first on the highest card, then on the next higher card, and finally on the third card.

If each has a flush, or if each has a hand of lower rank than one pair, the hand containing the highest card wins. The highest cards in the two hands being identical, the winner is determined by the rank of the next-highest card, and so on down to the lowest card in the two hands, if necessary.

When two players have hands which are in all respects identical, except for the suits of the cards, and when they are jointly highest in the showdown, they divide the pot as evenly as possible. In determining the rank of poker hands, the suits are meaningless.

Acknowledgements

Many thanks to Michael Frank for contributing enormously to the 80+ man-hours spent on this project so far. Thanks also to Lynn Stein, my adviser, for discouraging me from spending too much more time on it!