

# Creating Rules For President AI Players

Copyright ©1997 Lost Shaker Productions, Inc.

## Introduction

When we created President, we decided to move all of the decision-making processes of the artificial intelligence (AI) into text files, rather than code them into the game directly. The benefits of this are obvious - you don't need access to the program code in order to change the AI, you can have multiple distinct AI personalities, and it allows the users of the program (rather than just the creators) to create their own AI personalities. This document describes all that you need to know to create your own AI personalities.

An AI personality is driven by a set of rules of how to play. Each personality has one or more rules associated with it, arranged in a specific order, stored in text files, with an extension of .RUL. Each individual personality gets their own RUL file.

Each rule describes a possible play, followed by an optional list of conditions that must all be true. A RUL file will have rules in it that handle various circumstances, like what to do when you have the lead, or need to follow three of a kind, or when to play a two. Some of the rules won't be valid at various times. For example, a rule describing when to play a two assumes you actually have a two to play. If you don't, the rule is ignored. Likewise, if a rule describes when to play three of a kind, and the three of a kind you have is less than the three of a kind that are currently the upcards, or you don't have three of a kind, the rule will be ignored. **The thing to remember is that the only rules considered are the ones that apply to the current upcards, and that you are capable of playing (due to holding the right cards.)**

Rules are evaluated from the top of the file to the bottom. Once a rule has been found that applies (i.e., the player has the potential card, and the potential card is valid to play, and all of the conditions of the rule are true), then the rule is used and the card is played. If no rules are used, the player passes.

Rule files can contain blank lines and comments. Comment lines must begin with a semi-colon. Each rule must be on a line of it's own. Spaces and tabs will be ignored, and the case (upper or lower) of each rule is also ignored.

To use a RUL file, simply copy it into the President directory and run the game. Any errors in your rules will be caught and reported when the program first starts.

## Notation:

Standard programming notation will be used in this document. When you see something like <variable>, the thing inside of the <> characters is meant as a comment to you, rather than something that you should literally type. Instead of typing "<variable>", you would type something like "NumMyTwos". Also, square brackets mean that the thing inside them is optional, and may be left off.

## **Rule Structure:**

Each rule has the following structure:

<potential\_cards> [<optional\_list\_of\_conditions\_that\_must\_be\_met>]

Where <potential\_cards> can be one of the following:

<c> where <c> is 2, 3, 4, 5, 6, 7, 8, 9,  
T for a 10,  
J, Q, K, A,  
L for the lowest card you hold (not counting any twos)  
H for highest card you hold (not counting any twos), or  
P for pass.

For “H” and “L” only, the following modifiers can be added:

<c>s for a single card. (The “s” is optional. If you leave it off, it’s the same as “s”.)  
<c>p for a pair of the card  
<c>t for three of a kind of the card  
<c>q for four of a kind of the card  
<c>\* for the needed number of the kind of the card (to match the number of upcards. If you have the lead, it matches the lowest or highest card(s) in your hand.)

The potential cards and the conditions must be separated by blanks or tabs.

## **Potential Cards:**

The potential card part of a rule signifies the card or cards that could potentially be played, if they apply, and if the conditions that follow are all true. It is first looked at to determine if it applies to the current situation. For example, if your potential card is “LP”, which means the lowest pair you currently hold, but the current upcard is a single six, the rule will be ignored.

Note that “P” is a valid potential card. In this case, you’re not really considering playing a particular card - you’re considering passing. “P” is always valid, since you can always pass, no matter what the circumstance. In this case, the “P” rule will be played only if the conditions on the rule are satisfied.

The following is a set of examples of potential cards as would be used in a rule, along with a description of what they mean. Note that the player must have the card(s) in hand, and it must be a valid play in order for the rule to be considered. Also, all of the conditions of the rule must be met in order for the rule to be used.

2	a deuce	
P	pass, assuming the conditions are all true	
L	the lowest card in the player’s hand that can be played (not counting any twos)	
LP	the lowest pair in the player’s hand that can be played (not counting any twos)	
HQ	the highest four of a kind in the player’s hand that can be played (not counting any	twos)

L\* the lowest cards in your hand that are playable. If the upcards are a pair, then this would be the lowest playable pair you have, even if you have to split up three of a kind. If there are no upcards (i.e., you have the lead), then this would be the lowest card or cards you have.

Note that the number of cards needed (specified by the trailing “S”, “P”, “T”, or “Q”, means that the AI player must have exactly that number of cards. Let’s suppose that a pair of fours are the upcards, and in your RUL file you have a rule that starts with “LP”, and the players cards are “8 8 8 9 10 10 J”. The potential cards in this case will be the pair of tens, rather than the eights. Since you have three eights, it doesn’t match your requirement of a pair.

However, if you specify “L\*” instead of “LP”, the AI will search for the lowest pair (since that’s what’s needed, due to the pair of fours as the upcards), and will break up the triple eights you have.

L\* and H\* mean, therefore, that the AI should break any pairs, trips, or quads in order to get the absolute lowest or highest cards in the hand. If you have the lead, L\* and H\* mean “Play the lowest (or highest) cards in my hand”.

### **Conditions:**

Conditions are zero or more comparisons of a game variable to a value, separated by blanks or tabs. A game variable is a piece of information about the game, the current player, the cards currently held, or possibly the cards being considered for play. Each condition looks like this:

```
<GameVar> <Comparison> <Value>
```

Note that you must have blanks or tabs between the variable, comparison, and value. Here are some examples of rules that have conditions:

```
; Play a two if I have it, I’m not leading,  
; and I have more than one two in my hand.  
2      Lead = 0      TotMyTwos > 1  
  
; Play a two if I have only it and one other type of card  
2      NumDiffVals = 1  
  
; Play my lowest card as long as I’m not splitting a pair up  
L      NumThisValue = 1
```

There is a limit of ten conditions per rule.

### **Game Variables:**

The following are all of the game variables currently available for conditions. Note that you can use the short variable name to save typing, but the full variable name increases readability of your RUL files. Also, you don’t have to type the entire variable name. If you wish, use the first few letters of the variable. When you use the AI Debug screen in the game, all of your variables will be displayed with their full variable names, even if you’ve typed them in with the short names.

Short Variable Name	Full Variable Name	Meaning	Possible Values
L	Lead	Do I have the lead?	0 = No 1 = Yes
P	Rank	What is my position (rank)? Note that this is not valid for game 1, since no player has a rank at that point.	1 = President 2 = VP 3 = Neutral 4 = Vice Loser 5 = Loser 6 = N/A
V	NumDiffVals	How many different values do I hold (not counting twos)?	0 - 10
T	TwosLeft	How many twos are left to be played against me? That is, not counting my twos, and the twos that have been played, how many are left?	0 - 4
C	CardVal	What is the value of the card I am considering? (This would be used with the lowest/highest card indicator, to find out exactly the value I'm considering)	3 - 10, J (or 11), Q (or 12), K (or 13), A (or 14), 2 (or 15)
R	Random	A random value, generated each time the variable is referenced.	1 - 100
%	%Played	Percent of the 50 cards that have been played. You can use this as an indicator of how much of the game has been played.	1 - 100
N	HighestOppCards	What is the highest number of cards held by any of my opponents?	1 - 10
S	NumThisValue	How many cards of this value do I hold?	1 - 4
O	FollowOdds	What is the percentage chance of a player being able to follow the potential card(s)?	0 - 100
H	HigherPlayers	How many players of higher rank are still in the game?	0 - 4
F	PlayersLeft	How many players in total are still in the game?	1 - 5
I	TotMyCards	How many cards do I hold?	1 - 10
G	TotMyTwos	How many twos do I hold?	0 - 4
A	HandValue	What is the average card value of my hand, not counting any twos?	3 - A (14)
M	MatchesUpCard	Does the card you are considering match the upcard?	0 = No 1 = Yes
W	MyLowestCard	What is my lowest card value, not counting twos?	3 - 14
E	MyHighestCard	What is my highest card value, not counting twos?	3 - 14

### **Comparisons:**

Comparisons are simply > (greater than), < (less than), = (equals to), or ! (not equals to), followed by a value.

### **Values:**

Values are usually integer numbers, although they can be a single letter when comparing to a card value. When dealing with card values, the values are 3 - 10, 11 or J for Jack, 12 or Q for Queen, 13 or K for King, 14 or A for Ace, and 15 or 2 for a two. The reason that 15 can be used for a two is that it is the true numeric value of the card, in comparison to all of the other cards. By using these values, you can do valid less-than or greater-than comparisons.

### **Examples:**

; Play all twos to get the lead, if I only have one other value (i.e., I'm going out)

2 NumDiffVals = 1

; Otherwise, just play the lowest thing possible - but don't play a q/t/p higher than a jack

lq CardVal < Q

lt CardVal < Q

lp CardVal < Q

; Play the lowest non-paired card only if it's less than a queen

L CardVal < Q

; Otherwise, play the lowest card I've got - even if I split a pair, trip, or quad

L\*