Tic tac toe

The following function evaluates the winner of the game (or no winner/draw).

ganador <- function(juego, player) {  
 game <- matrix(juego, nrow = 3, byrow = T)  
 hor <- rowSums(game)  
 ver <- colSums(game)  
 diag <- c(sum(diag(game)), sum(diag(apply(game, 1, rev))))  
 if (-3 %in% c(hor, ver, diag))  
 return(-10)  
 if (3 %in% c(hor, ver, diag))  
 return(10)  
 else  
 return(0)  
}

The following code is the actual minimax algorithm. The comment lines are a way to make the algorithm faster. The first decision (when there are 9 options to play) takes a little while; however it always choses one of the corner options so I've simply randomized over the corner options.

The function outputs a list with the move (1-9) as the first element and the utility (0, 10 or -10) as the second element. The utility is only useful inside the function, to use the algorithm for a game simply use the $move element.

minimax <- function(juego, player) {  
 free <- which(juego == 0)  
 if(length(free) == 1) {  
 juego[free] <- player  
 return(list(move = free, U = ganador(juego, player)))  
 }  
 # if(length(free) == 9){  
 # ran <- sample(c(1, 3, 7, 9), 1)  
 # juego[ran] <- player  
 # return(list(move = ran, U = ganador(juego, player)))  
 # }  
 poss.results <- rep(0, 9)  
 for(i in free) {  
 game <- juego  
 game[i] <- player  
 poss.results[i] <- ganador(game, player)  
 }  
 mm <- ifelse(player == -1, "which.min", "which.max")  
 if(any(poss.results == (player\*10))) {  
 move <- do.call(mm, list(poss.results))  
 return(list(move = move, U = poss.results[move]))  
 }  
 for(i in free) {  
 game <- juego  
 game[i] <- player  
 poss.results[i] <- minimax(game, -player)$U  
 }  
 random <- runif(9, 0, 0.1)  
 poss.results[-free] <- 100 \* -player  
 poss.results <- poss.results + (player \* random)  
 move <- do.call(mm, list(poss.results))  
 return(list(move = move, U = poss.results[move]))  
}

I've drawn the board, made the human player and set up the game almost exactly as you've set it up on your blog. That only leaves including these functions and trying it out for yourself.

Alberto

P.S. Sorry some words in the code are in spanish. I'm a native spanish speaker.