

```
#!/usr/bin/perl
#
#
#A very basic tic-tac-toe program (the computer chooses randomly)
#

use strict;
use CGI;
use Socket;

my ($rounds, $round_temp, $squares, $page, $x, $y, $z, $cell,
    $player_move, @available_choices, $computer_move, @choices, $round,
    $winner, $player_move_pretty, $computer_move_pretty);

my ($round_minus_one); #bug fix (was recording moves for round1 as round2

$page = CGI->new();

print $page->header;
print $page->start_html();

# print table beginnings

print("<table width=\"90\%\" border=0 cellpadding=15>\n");
print("<tr valign=middle>\n");

# left cell is tic tac toe table

print("<td align=center>\n");

#
# find out which round it is so we know how to define $squares
#

unless ($page->param('round')) {
    $round = 0;
}else {
    $round = $page->param('round');
}

#
# set array of tic tac toe squares
#

for my $x (0..2) {
    for my $y (0..2) {
        $squares->[$x][$y] = $page->param("$x$y");
    }
}

#
# set array for determing history of moves (recorded by round)
#

if ($round > 0) {
    for my $rn (1..5) {
        $rounds->{"round$rn"} =
            { player => $page->param("round${rn}_x"),
              computer => $page->param("round${rn}_o"),
            }
    }
}

#
```

```

# increment $round to give it a new hidden value
#

$round = $round + 1;
$round_minus_one = $round - 1;
print ("Round is: $round<br>\n");

##
## get player move using subroutine (subroutine stores it to $squares array
)
##

$player_move = $page->param('choice');
if ($player_move) {
    $round_temp = "round" . $round_minus_one;
    player_moves($player_move, $page, $squares);
    $player_move_pretty = make_move_pretty($player_move);
    $rounds->{$round_temp}->{'player'} = $player_move_pretty;

#
# evaluate for winner after player moves
#

    $winner = evaluate_board($squares);
    if ($winner eq "x") {
        print ("<font color='blue'>Player Won!</font><p>\n");
        print_table($squares, $page, $winner, $round, $rounds, "final");
    }else {

#
# check to see if player won. if player won, don't do the rest of this

#
# get available choices for computer choices
#

        @available_choices = get_available_choices($squares);

#
# get computer move
#

        $computer_move = $available_choices[int rand @available_choices];
        ($x, $y) = split(/,/, $computer_move);
        #get coordinates for computer move and store in $x and $y
        $squares->[$x][$y] = "o"; ## change square to "o"
        $round_temp = "round" . $round_minus_one;
        $computer_move_pretty = make_move_pretty($computer_move);
        $rounds->{$round_temp}->{'computer'} = $computer_move_pretty;

    } #matches else {
} #matches if ($player_move)

#
# now that we have the array with computer and player choices, see if there
is a winner
#
$winner = evaluate_board($squares);
if ($winner eq "o") { #we already checked for x before
    print ("<font color = 'blue'>Computer Won!</font><p>\n");
    print_table($squares, $page, $winner, $round, $rounds, "final");
}

if (($winner ne "o") and ($winner ne "x")) {

```

```

    print ("<font color='blue'>No winner yet</font><p>\n");

print_table($squares, $page, $winner, $round, $rounds);

}

# end table cell

print ("</td>\n");
print ("<td align=middle>\n");

# get printable versions of moves and print choices

if (($player_move) or ($computer_move)) {
    foreach $x(1..$round_minus_one) {
        $round_temp = "round" . $x;
        print ("<p><b>Round $x:</b><br>\n");
        print ("\tplayer: " . $rounds->{$round_temp}->{'player'} . "<br>\n");
        print ("\tcomputer: " . $rounds->{$round_temp}->{'computer'} . "<br>\n");
    }
} else {
    print ("No moves yet.\n");
}

# end table

print ("</td></tr></table>\n");

print $page->end_html();

sub player_moves {
    my $move = $_[0]; ##get move
    my $page = $_[1]; ## import page object
    my $squares = $_[2]; ## import array so we can change square
    my ($x, $y, $z);

    foreach $x(0..2) {
        foreach $y(0..2) { ##test for each square
            if ($move eq "$x$y") {
                $squares->[$x][$y] = "x"; ## define array element for choice
            }
        }
    }
}

sub get_available_choices {
    my $squares = $_[0];
    my ($x, $y, $z);
    my @available_choices = ();

    foreach $x(0..2) {
        foreach $y(0..2) {
            unless (($squares->[$x][$y] eq "x") or ($squares->[$x][$y] eq "o"))
        ) {
            $z = "$x$y";
            push (@available_choices, $z);
        }
    }
}
return @available_choices;
}

```

```

sub print_hidden_values {
    my $page = $_[0];
    my $squares = $_[1];
    my $rounds = $_[2];
    my ($x, $y, $cell, $round);

    #print hidden values for cells
    foreach $x(0..2) {
        foreach $y(0..2) {
            $cell = "$x$y";
            print("<input type='hidden' name='$cell' value='" . $squares->[
            $x][$y] . "'>");
            print("\n");
        }
    }

    #print hidden values for rounds (history)
    foreach $x(1..5) {
        $round = "round" . $x;
        print("<input type='hidden' name='$round" . "_x' value='" . $rounds
        ->{$round}->{'player'} . "'>\n");
        print("<input type='hidden' name='$round" . "_o' value='" . $rounds-
        >{$round}->{'computer'} . "'>\n");
    }
}

sub evaluate_board {
    my ($board) = $_[0];

    my @table = (
        [ 0,0 , 0,1 , 0,2 ],
        [ 1,0 , 1,1 , 1,2 ],
        [ 2,0 , 2,1 , 2,2 ],
        [ 0,0 , 1,0 , 2,0 ],
        [ 0,1 , 1,1 , 2,1 ],
        [ 0,2 , 1,2 , 2,2 ],
        [ 0,0 , 1,1 , 2,2 ],
        [ 0,2 , 1,1 , 2,0 ],
    );

    for my $win (@table) {
        my ($x1, $y1, $x2, $y2, $x3, $y3) = @$win;
        if ($board->[$x1][$y1] eq $board->[$x2][$y2]
            && $board->[$x1][$y1] eq $board->[$x3][$y3]) {
            return $board->[$x1][$y1];
        }
    }
    return;
}

sub print_table {
    my ($squares, $page, $winner, $round, $rounds, $final) = @_;
    my ($visitor, $visitor_name, $time);

    ## print ending table

    print $page->startform(-method=>'POST');

    print_hidden_values($page,$squares,$rounds);

    print("<input type='hidden' name='round', value='$round'>\n");

    print("<table border=1 cellpadding=10>\n");
    print("<tr valign=middle>\n");

```

```

    for my $row (0..2) {
        for my $col (0..2) {
            my $cell = $squares->[$row][$col];
            if ($cell eq "") {
                $cell = $final ? "?"
                    : "<td><input type='checkbox' name='choice' value='$row$col
'></td>\n";
            }
            print ("<td align=center>" . $cell . "</td>\n");
        }
        print ("</tr><tr valign=middle>\n") unless $row == 2;
    }
    print ("</tr></table><p>\n");

    if ($final) {
        print ("<font color='red'>Note: if you pick more than one square, y
our choice will be the upper and leftmost square that you choose!!</font><p>
\n"\);
    }

    print \$page->endform\(\);

    print \$page->end\_html\(\);

    if \(\$final\) {
        #print log of play

        \$visitor = \$page->remote\_host\(\);
        if \(\$visitor =~ /\d\*\.\d\*\.\d\*\.\d\*/\) {
            \$visitor\_name = gethostbyaddr\(inet\_aton\(\$visitor\), AF\_INET\);
        }

        \$time = localtime\(time\(\)\);

        # open \(MAIL, "| /usr/sbin/sendmail -t"\);
        # print MAIL "To: author@example.com\n";
        # print MAIL "Subject: tic tac toe results\n";
        # print MAIL "\n\$visitor, \$visitor\_name: \$time: \$winner on round \$ro
und";
        # close MAIL;
    }
}

sub make\_move\_pretty {
    my %squares\_names = \("00" => "top left",
                        "01" => "top center",
                        "02" => "top right",
                        "10" => "center left",
                        "11" => "center",
                        "12" => "center right",
                        "20" => "lower left",
                        "21" => "lower center",
                        "22" => "lower right"
    \);

    return \$squares\_names{\$\_\[0\]};
}

```