```
 1     #!/usr/bin/perl
 2     #
 3
 4     #
 5     #A very basic tic−tac−toe program (the computer chooses randomly)
 6     #
 7
 8
 9     use strict;
10     use CGI;
11     use Socket;
12
13     my ($rounds, $round_temp, $squares, $page, $x, $y, $z, $cell,
14         $player_move, @available_choices, $computer_move, @choices, $round,
15         $winner, $player_move_pretty, $computer_move_pretty);
16
17     my ($round_minus_one);  #bug fix (was recording moves for round1 as roun
d2
18
19     $page = CGI−>new();
20
21     print $page−>header;
22     print $page−>start_html();
23
24     # print table beginnings
25
26     print ("<table width=\"90\%\" border=0 cellpadding=15>\n");
27     print ("<tr valign=middle>\n");
28
29     # left cell is tic tac toe table
30
31     print ("<td align=center>\n");
32
33     #
34     # find out which round it is so we know how to define $squares
35     #
36
37     unless ($page−>param('round')) {
38        $round = 0;
39     }else {
40        $round = $page−>param('round');
41     }
42
43     #
44     # set array of tic tac toe squares
45     #
46
47     for my $x (0..2) {
48       for my $y (0..2) {
49         $squares−>[$x][$y] = $page−>param("$x$y");
50       }
51     }
52
53     #
54     # set array for determing history of moves (recorded by round)
55     #
56
57     if ($round > 0) {
58             for my $rn (1..5) {
59               $rounds−>{"round$rn"} =
60                 { player   => $page−>param("round${rn}_x"),
61                   computer => $page−>param("round${rn}_o"),
62                 }
63             }
64     }
65
```

```
    66     #
    67     # increment $round to give it a new hidden value
    68     #
    69
    70     $round = $round + 1;
    71     $round_minus_one = $round − 1;
    72     print ("Round is: $round<br>\n");
    73
    74     ##
    75     ## get player move using subroutine (subroutine stores it to $squares ar
ray)
    76     ##
    77
    78     $player_move = $page−>param('choice');
    79     if ($player_move) {
    80        $round_temp = "round" . $round_minus_one;
    81        player_moves($player_move, $page, $squares);
    82        $player_move_pretty = make_move_pretty($player_move);
    83        $rounds−>{$round_temp}−>{'player'} = $player_move_pretty;
    84
    85     #
    86     # evaluate for winner after player moves
    87     #
    88
    89        $winner = evaluate_board($squares);
    90        if ($winner eq "x") {
    91           print ("<font color='blue'>Player Won!</font><p>\n");
    92           print_table($squares, $page, $winner, $round, $rounds, "final");
    93        }else {
    94
    95     #
    96     # check to see if player won. if player won, don't do the rest of this
    97
    98
    99     #
   100     # get available choices for computer choices
   101     #
   102
   103           @available_choices = get_available_choices($squares);
   104
   105     #
   106     # get computer move
   107     #
   108           $computer_move = get_computer_choice(@available_choices);
   109           ($x, $y) = split(//, $computer_move);
   110              #get coordinates for computer move and store in $x and $y
   111           $squares−>[$x][$y] = "o";   ## change square to "o"
   112           $round_temp = "round" . $round_minus_one;
   113           $computer_move_pretty = make_move_pretty($computer_move);
   114           $rounds−>{$round_temp}−>{'computer'} = $computer_move_pretty;
   115
   116        } #matches else {
   117
   118     } #matches if ($player_move)
   119
   120
   121     #
   122     # now that we have the array with computer and player choices, see if th
ere is a winner
   123     #
   124     $winner = evaluate_board($squares);
   125     if ($winner eq "o") {  #we already checked for x before
   126        print ("<font color = 'blue'>Computer Won!</font><p>\n");
   127        print_table($squares, $page, $winner, $round, $rounds, "final");
   128     }
   129
```

```
   130     if (($winner ne "o") and ($winner ne "x")) {
   131        print ("<font color='blue'>No winner yet</font><p>\n");
   132
   133
   134     print_table($squares, $page, $winner, $round, $rounds);
   135
   136
   137     }
   138
   139     # end table cell
   140
   141     print ("</td>\n");
   142     print ("<td align=middle>\n");
   143
   144     # get printable versions of moves and print choices
   145
   146     if (($player_move) or ($computer_move)) {
   147        foreach $x(1..$round_minus_one) {
   148           $round_temp = "round" . $x;
   149           print ("<p><b>Round $x:</b><br>\n");
   150           print ("\tplayer: " . $rounds->{$round_temp}->{'player'} . "<br>\n
");
   151           print ("\tcomputer: " . $rounds->{$round_temp}->{'computer'} . "<b
r>\n");
   152        }
   153     }else {
   154        print ("No moves yet.\n");
   155     }
   156
   157     # end table
   158
   159     print ("</td></tr></table>\n");
   160
   161     print $page->end_html();
   162
   163
   164     sub player_moves {
   165        my $move = $_[0];  ##get move
   166        my $page = $_[1];  ## import page object
   167        my $squares = $_[2];  ## import array so we can change square
   168        my ($x, $y, $z);
   169
   170        foreach $x(0..2) {
   171           foreach $y(0..2) {         ##test for each square
   172              if ($move eq "$x$y") {
   173                 $squares->[$x][$y] = "x";   ## define array element for choi
ce
   174              }
   175           }
   176        }
   177     }
   178
   179     sub get_available_choices {
   180        my $squares = $_[0];
   181        my ($x, $y, $z);
   182        my @available_choices = ();
   183
   184        foreach $x(0..2) {
   185           foreach $y(0..2) {
   186              unless (($squares->[$x][$y] eq "x") or ($squares->[$x][$y] eq "
o")) {
   187               $z = "$x$y";
   188               push (@available_choices, $z);
   189              }
   190           }
   191        }
```

```
   192        return @available_choices;
   193     }
   194
   195     sub get_computer_choice {
   196        my @available_choices = @_;
   197        my $length = @available_choices;
   198        my $number = int(rand() * ($length − 1));
   199        my $choice = $available_choices[$number];
   200        return $choice;  ## this will be a coordinate, in the form of $x$y fr
om $z above
   201     }
   202
   203
   204     sub print_hidden_values {
   205        my $page = $_[0];
   206        my $squares = $_[1];
   207        my $rounds = $_[2];
   208        my ($x, $y, $cell, $round);
   209
   210        #print hidden values for cells
   211        foreach $x(0..2) {
   212          foreach $y(0..2) {
   213                $cell = "$x$y";
   214              print ("<input type=’hidden’ name=’$cell’ value=’" . $squares−>
[$x][$y] . "’>");
   215                print ("\n");
   216             }
   217        }
   218
   219        #print hidden values for rounds (history)
   220        foreach $x(1..5) {
   221           $round = "round" . $x;
   222           print ("<input type=’hidden’ name=’$round" . "_x’ value =’" . $rou
nds−>{$round}−>{’player’} . "’>\n");
   223           print ("<input type=’hidden’ name=’$round" . "_o’ value=’" . $roun
ds−>{$round}−>{’computer’} . "’>\n");
   224        }
   225     }
   226
   227
   228     sub evaluate_board {
   229         my ($board) = $_[0];
   230
   231         my @table = (
   232            [ 0,0 , 0,1 , 0,2 ],
   233            [ 1,0 , 1,1 , 1,2 ],
   234            [ 2,0 , 2,1 , 2,2 ],
   235            [ 0,0 , 1,0 , 2,0 ],
   236            [ 0,1 , 1,1 , 2,1 ],
   237            [ 0,2 , 1,2 , 2,2 ],
   238            [ 0,0 , 1,1 , 2,2 ],
   239            [ 0,2 , 1,1 , 2,0 ],
   240         );
   241
   242         for my $win (@table) {
   243           my ($x1, $y1, $x2, $y2, $x3, $y3) = @$win;
   244           if ($board−>[$x1][$y1] eq $board−>[$x2][$y2]
   245               &&  $board−>[$x1][$y1] eq $board−>[$x3][$y3]) {
   246             return $board−>[$x1][$y1];
   247           }
   248         }
   249         return;
   250     }
   251
   252     sub print_table {
   253        my ($squares, $page, $winner, $round, $rounds, $final) = @_;
```

```
   254        my ($visitor, $visitor_name, $time);
   255
   256        ## print ending table
   257
   258        print $page->startform(-method=>'POST');
   259
   260        print_hidden_values($page,$squares,$rounds);
   261
   262        print ("<input type='hidden' name='round', value='$round'>\n");
   263
   264        print ("<table border=1 cellpadding=10>\n");
   265        print ("<tr valign=middle>\n");
   266        for my $row (0..2) {
   267          for my $col (0..2) {
   268             my $cell = $squares->[$row][$col];
   269             if ($cell eq "") {
   270               $cell = $final ? "?"
   271                     : "<td><input type='checkbox' name='choice' value='$row$
col'></td>\n";
   272             }
   273             print ("<td align=center>" . $cell . "</td>\n");
   274          }
   275          print ("</tr><tr valign=middle>\n") unless $row == 2;
   276        }
   277        print ("</tr></table><p>\n");
   278
   279        if ($final) {
   280          print ("<a href=\"http://www.example.com/cgi-bin/tic_tac.cgi\">Play
 Again!</a>") ;
   281        } else {
   282          print $page->submit();
   283
   284          print ("<p><font color='red'>Note: if you pick more than one square
, your choice will be the upper and leftmost square that you choose!!</font><p>\n")
;
   285        }
   286
   287        print $page->endform();
   288
   289
   290        print $page->end_html();
   291
   292        if ($final) {
   293          #print log of play
   294
   295          $visitor = $page->remote_host();
   296          if ($visitor =~ /\d*\.\d*\.\d*\.\d*/) {
   297            $visitor_name = gethostbyaddr(inet_aton($visitor), AF_INET);
   298          }
   299
   300          $time = localtime(time());
   301
   302          #   open (MAIL, "| /usr/sbin/sendmail -t");
   303          #   print MAIL "To: author\@example.com\n";
   304          #   print MAIL "Subject: tic tac toe results\n";
   305          #   print MAIL "\n$visitor, $visitor_name: $time: $winner on round
$round";
   306          #   close MAIL;
   307        }
   308
   309     }
   310
   311     sub make_move_pretty {
   312        my %squares_names = ("00" => "top left",
   313                             "01" => "top center",
   314                             "02" => "top right",
```

```
315                                 "10" => "center left",
316                                 "11" => "center",
317                                 "12" => "center right",
318                                 "20" => "lower left",
319                                 "21" => "lower center",
320                                 "22" => "lower right"
321                                );
322
323         return $squares_names{$_[0]};
324     }
325
326
```