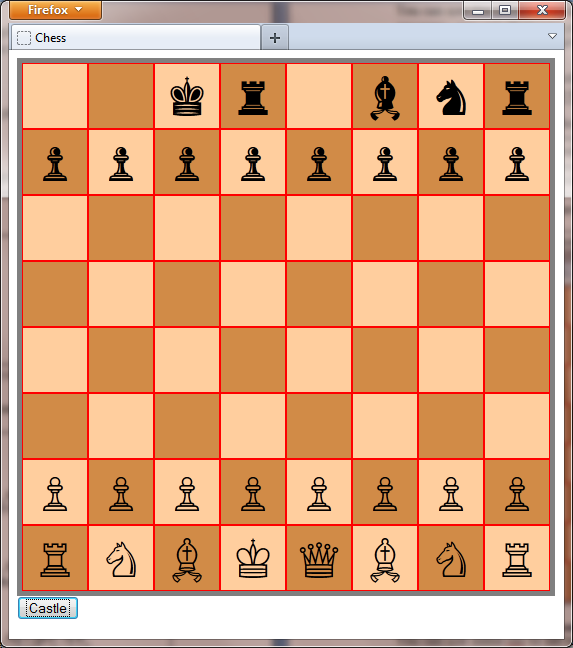
# jQuery Tutorial – Create a Chess Board

In this tutorial you will be using jQuery to create an application where users can move element around. You will practice using arrays to record where pieces are and restrict user actions.

## Final Result

You can move pieces around the board and take other pieces.



## Task One – Set up the board

You must create a square board. It should be big enough to contain an 8 by 8 grid of squares. Each square has a one pixel border so the length of each side must be 66x8 = 528 pixels.

Download <http://progzoo.net/chess.zip> Add the following to chess.css

|  |
| --- |
| #board{  width:528px;  height:528px;  border:solid thick gray;  position:relative;  padding:0px;  } |

You will need to add the 64 squares using jQuery. They should fit exactly so that none of the red background is visible.

The following jQuery creates some of the squares you need. Perhaps you could use loops to save yourself some effort:

|  |
| --- |
| $(function(){  var sq;  sq = $('<div/>');  sq.css({position:'absolute',width:64,height:64,  left:1,top:1,background:'#ffce9e'});  $('#board').append(sq);  sq = $('<div/>');  sq.css({position:'absolute',width:64,height:64,  left:67,top:1,background:'#d18b47'});  $('#board').append(sq);  sq = $('<div/>');  sq.css({position:'absolute',width:64,height:64,  left:133,top:1,background:'#ffce9e'});  $('#board').append(sq);  sq = $('<div/>');  sq.css({position:'absolute',width:64,height:64,  left:1,top:67,background:'#d18b47'});  $('#board').append(sq);  }); |

## Task Two – Add some pieces

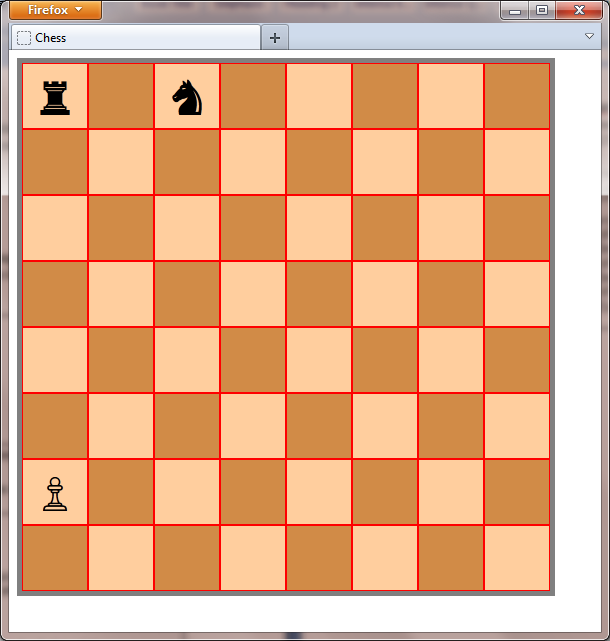
You can use the phrase **var sqlist = $('#board div');** to get a list of all of the squares – you can then add pieces to the right places.

A black rook has Unicode &#9820; the black knight is &#9822; - you can find the complete list at Wikipedia:

http://en.wikipedia.org/wiki/Chess\_symbols\_in\_Unicode

You can use this code to put the first few pieces on the board:

|  |
| --- |
| var sqlist = $('#board>div');  var br1 = $('<div></div>').html('&#9820;');  $(sqlist[0]).append(br1);  var bn1 = $('<div></div>').html('&#9822;');  $(sqlist[2]).append(bn1);  var wp1 = $('<div></div>').html('&#9817;');  $(sqlist[48]).append(wp1); |



Your board should look something like this:

You can use some css to set text-align to center and font-size to 300%

You might be able to use the following list and a loop to put all of the pieces on:

|  |
| --- |
| var pieceList = [9820,9822,9821,9819,9818,9821,9822,9823,  9823,9823,9823,9823,9823,9823,9823,9823,  0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,  0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,  9817,9817,9817,9817,9817,9817,9817,9817,  9814,9816,9815,9812,9813,9815,9816,9814  ]; |

## Task Three – allow dragging

You should have a full chess board. Allowing users to drag pieces is just one line of code. Each piece is a div, inside a square div, inside the board div. You apply the function draggable (from jQuery UI).

|  |
| --- |
| $('#board div div').draggable(); |

You will find some problems:

* You can see the white pieces when you drag them up the board but not the black pieces when you drag them down.
* The user sees the default **select** cursor when the mouse is over a draggable item.

To solve these problems:

* Use css to set the **z-index** of the pieces
* Use css to set the **cursor** property

## Task Four – handling the drop

The drop event can be handled. To do this you call **droppable** on the squares – these are the direct descendants of **#board** – addressed as $('#board>div')

The following code will remove both the piece that was dropped and the square that it was dropped on:

|  |
| --- |
| $('#board>div').droppable({drop:function(event,ui){  ui.draggable.hide('slow');  $(this).hide('slow');  }}); |

You want different action to take place:

* Append the dragged item to the square that it is dropped on.
* You may need to set left and top of the dragged item to 0 – otherwise the offset will accumulate.
* The square that it got dropped on should be made empty – you should do this *before* you add the new piece.

## Task Five - Restricting movement

Pawns can only move up and down (we will not deal with the diagonal take).

* Look up the documentation on jquery-ui draggable and find out how to **constrain** the drag.

The following code will constrain the drag for the first black pawn:

|  |
| --- |
| $(sqlist[8]).children().draggable({axis:'x'}); |

## Task Six – Implementing a castle move

Add a button to make the black king castle. The following should take place:

* Locations 1, 2 and 3 must be emptied
* The piece in location 0 (the black castle) moves to location 3
* The piece in location 4 (the black king) moves to location 1

Help:

You can add a button using code such as:

|  |
| --- |
| var btn = $('<button/>',{text:'Click here'});  btn.click(function(){  alert('Yes?');  });  $('body').append(btn); |

You can make the piece in location 1 disappear with code such as:

|  |
| --- |
| $(sqlist[1]).children().hide('slow'); |

You can move the piece from location 0 to location 20 using this code:

|  |
| --- |
| $(sqlist[20]).append($(sqlist[0]).children()); |

You might try to animate the move. The following is not right – but it moves something somewhere:

|  |
| --- |
| $(sqlist[8]).children().animate({top:'+='+66\*4}); |

## Using Images instead of UNICODE

You can use the images in the file pieces.png

The following code will put the rook (in column 2 row 1 in pieces.png)

It will put the knight (in column 4 row 1 in pieces.png)

Notice that in counting rows and columns we start at zero.

|  |
| --- |
| var sl = $('#board div');  var br1 = $('<div/>');  br1.css({width:64,height:64,  background:'url(pieces.png)',  backgroundPosition:'-'+(62\*2+4)+'px'+  ' -'+(62\*1+4)+'px'});  $(sl[0]).append(br1);  var bn1 = $('<div/>');  bn1.css({width:64,height:64,  background:'url(pieces.png)',  backgroundPosition:'-'+(62\*4+4)+'px'+  ' -'+(62\*1+4)+'px'});  $(sl[1]).append(bn1); |

Answers

Task One – Set up the board:

|  |
| --- |
| $(function(){  for (var i=0;i<8;i++){  for(var j=0;j<8;j++){  var sq;  sq = $('<div/>');  sq.css({position:'absolute',width:64,height:64,  left:1+66\*j,top:1+66\*i});  if ((i+j)%2==0)  sq.css({background:'#ffce9e'});  else  sq.css({background:'#d18b47'});  $('#board').append(sq);  }  }  }); |

Task Two – Put the pieces on

|  |
| --- |
| $(function(){  // As Above    var sqlist = $('#board>div');  var pieceList=[9820,9822,9821,9819,9818,9821,9822,9820,  9823,9823,9823,9823,9823,9823,9823,9823,  0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,  0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,  9817,9817,9817,9817,9817,9817,9817,9817,  9814,9816,9815,9812,9813,9815,9816,9814  ];  for(var i=0;i<pieceList.length;i++){  if (pieceList[i]==0) continue;  var p = $('<div></div>').html('&#'+pieceList[i]+';');  $(sqlist[i]).append(p);  }  $('#board div div').css({fontSize:'300%',  textAlign:'center'});  }); |

Task Three – Allow drag:

|  |
| --- |
| $('#board div div').css({fontSize:'300%',zIndex:100,  textAlign:'center'});  $('#board div div').draggable(); |

Task Four – handling the drop

|  |
| --- |
| $('#board>div').droppable({drop:function(event,ui){  $(this).children().hide('slow');  ui.draggable.css({left:0,top:0});  $(this).append(ui.draggable);  }}); |

Task Five – restrict pawns

|  |
| --- |
| for (var i=0;i<8;i++){  $(sqlist[8+i]).children().draggable({axis:'y'});  $(sqlist[48+i]).children().draggable({axis:'y'});  } |

Task Six – Castle

|  |
| --- |
| var btn = $('<button/>',{text:'Castle'});  btn.click(function(){  $(sqlist[1]).children().hide('slow');  $(sqlist[2]).children().hide('slow');  $(sqlist[3]).children().hide('slow');  $(sqlist[3]).prepend($(sqlist[0]).children());  $(sqlist[0]).append($(sqlist[4]).children());  });  $('body').append(btn); |

Task Six – Castle with animation

|  |
| --- |
| var btn = $('<button/>',{text:'Castle'});  btn.click(function(){  $(sqlist[1]).children().hide('slow');  $(sqlist[2]).children().hide('slow');  $(sqlist[3]).children().hide('slow');  var king = $(sqlist[4]).children();  var rook = $(sqlist[0]).children();  king.animate({left:'-='+66\*2});  rook.animate({left:'+='+66\*3});  });  $('body').append(btn); |