Coder Dojo 2012-03-10

More Shuffle Puzzle

LAST week...

display and starting to shuffle.
But you covered a lot of topics...

github.com/willknott github.com/coderdojo- I'm working on it

Last week's slides, code (puzzle3) and images are available

Huh? You created these variables...

currentPiece = null;
currentDropPiece = null;
Null means "No value".
mouse = {x:0,y:0};
This is a full one value array...and an object.

Think about it, what does a mouse position consist of...

pieces = [];
This is an empty array

Met arrays

http://www.w3schools.com/js/js_obj_array.asp

An array is a variable that can contain more than one value.

You can declare a complete array, or, push values on to an array.

Covered For loops

We are about to hit a for loop

http://www.w3schools.com/js/js_loop_for.asp

```
for (variable=startvalue;
       variable<=endvalue;
       variable=variable+increment)
      {
         do something
      }
```

And finished with a that function handles when we click on a piece in the puzzle.

Or rather... where.

```
function onPuzzleClick(e){
        if(e.layerX || e.layerX == 0){
            mouse.x = e.layerX - canvas.offsetLeft;
            mouse.y = e.layerY - canvas.offsetTop;
        }
        else if(e.offsetX || e.offsetX == 0){
            mouse.x = e.offsetX - canvas.offsetLeft;
            mouse.y = e.offsetY - canvas.offsetTop;
        }
        currentPiece = checkPieceClicked();
        // call the detection function
```

Nasty isn't it

This is needed because I don't know which browser you are using.
Some use Layer, some use Offset.

You want the coordinates inside the entire puzzle

Continued ....

```
if(currentPiece != null){
            stage.clearRect(currentPiece.xPos, currentPiece.yPos,
pieceWidth, pieceHeight);
            stage.save();
            stage.globalAlpha = .9;
            stage.drawImage(img, currentPiece.
sx,                         currentPiece.sy, pieceWidth, pieceHeight,
   mouse.x - (pieceWidth / 2), mouse.y - (pieceHeight /
2),                   pieceWidth, pieceHeight);
            stage.restore();
            document.onmousemove = updatePuzzle;
            document.onmouseup = pieceDropped;
        }
    }
```

?

if(currentPiece != null)

If your function currentPiece returns something...
stage.clearRect(currentPiece.xPos, currentPiece.yPos, pieceWidth, pieceHeight);

Delete the select piece

```
               stage.save();
```

Don't loose anything!!!

.

```
                stage.globalAlpha = .9;
```

Alpha means transparency, but its the transparency of the piece.

```
stage.drawImage(img, currentPiece.sx,                              currentPiece.
sy, pieceWidth, pieceHeight,
    mouse.x - (pieceWidth / 2), mouse.y - (pieceHeight /
2),                          pieceWidth, pieceHeight);
```

Draw the piece nudged a bit

```
                stage.restore();
```

And put everything back

then...

```
document.onmousemove = updatePuzzle;
document.onmouseup = pieceDropped;
```

More mouse functions but its calling functions we haven't done yet.
Save and reload and you'll see everything nudged a bit

This next function is **enormous**. It's doing loads of different things which I'll try my best to explain.

```
function updatePuzzle(e){
          currentDropPiece = null;
          if(e.layerX || e.layerX == 0){
              mouse.x = e.layerX - canvas.offsetLeft;
              mouse.y = e.layerY - canvas.offsetTop;
          }
          else if(e.offsetX || e.offsetX == 0){
             mouse.x = e.offsetX - canvas.offsetLeft;
             mouse.y = e.offsetY - canvas.offsetTop;
          }
         stage.clearRect(0,0, puzzleWidth, puzzleHeight);
```

meaning

```
currentDropPiece = null;
```
set stuff up
```
            if(e.layerX || e.layerX == 0){...........
            else if(e.offsetX || e.offsetX == 0){.........
```
We've seen this a few slides earlier

```
           stage.clearRect(0,0, puzzleWidth, puzzleHeight);
```
empty for redrawing

Continued.......

```
          var i;
          var piece;
//Loop through all the pieces
          for(i = 0;  i < pieces.length;  i++){
             piece = pieces[i];
             if(piece == currentPiece){  //If its the selected piece
                continue;
             }
             stage.drawImage(img,
                 piece.sx, piece.sy, pieceWidth, pieceHeight,
                piece.xPos, piece.yPos, pieceWidth, pieceHeight);
//Draw the piece in place
             stage.strokeRect(piece.xPos, piece.
yPos,                                                   pieceWidth, pieceHeight);
//Draw a rectangle around it
if(currentDropPiece == null){
                if(mouse.x < piece.xPos || mouse.x > (piece.xPos +
pieceWidth) || mouse.y < piece.yPos || mouse.y > (piece.yPos +
pieceHeight)){
                    //I'm not hovering over this piece, try the next
                }
                else{
                    currentDropPiece = piece;
                    stage.save();
                    stage.globalAlpha = .4;
                    stage.fillStyle = PUZZLE_HOVER_TINT;
                    stage.fillRect(currentDropPiece.xPos,
yPos,
                            pieceWidth, pieceHeight);
                    stage.restore();
                }
            }
        }
```

Hurray!!! The gigantic function is finally finished. Next we need to write a function that checks if we moved the piece from one point to another.

```
 function pieceDropped(e){
          document.onmousemove = null;
          document.onmouseup = null;
          if(currentDropPiece != null){
              var tmp = {xPos:currentPiece.xPos,yPos:currentPiece.
yPos};
              currentPiece.xPos = currentDropPiece.xPos;
              currentPiece.yPos = currentDropPiece.yPos;
              currentDropPiece.xPos = tmp.xPos;
              currentDropPiece.yPos = tmp.yPos;
          }
          resetPuzzleAndCheckWin();
      }
```

```
// 2 functions to go!!!! This one checks if we've won the game.
function resetPuzzleAndCheckWin(){
          stage.clearRect(0,0,puzzleWidth,puzzleHeight);
          var gameWin = true;
          var i;
          var piece;
          for(i = 0;i < pieces.length;i++){
              piece = pieces[i];
              stage.drawImage(img, piece.sx, piece.sy, pieceWidth,
pieceHeight, piece.xPos, piece.yPos, pieceWidth, pieceHeight);
              stage.strokeRect(piece.xPos, piece.yPos, pieceWidth,
pieceHeight);
              if(piece.xPos != piece.sx || piece.yPos != piece.sy){
                  gameWin = false;
              }
          }
```

Continued.....

```
    if(gameWin){
                setTimeout(gameOver,500);
            }
        }
```

The piece of code on this slide says that if we have won, wait 1/2 a second and then run the gameOver function which we do next.

//The gameOver function resets some of the event listeners and //gets the puzzle ready to play again.

```
 function gameOver(){
           document.onmousedown = null;
           document.onmousemove = null;
           document.onmouseup = null;
           initPuzzle();
       }
```

Now, we've got a nice game that we can play :-)
How can we change how many pieces are in the puzzle?
What about using a different image?