

Homework 2

Lane Kuhlman

Thu, Oct 2, 2008 at 6:32 PM

Lane Kuhlman – Homework #2

If I were to try and tell a friend how to construct a camel with the tangram, I would start by showing them the image of what the final camel's outline should look like. This could be a template that they align the pieces to. Then I would give them instruction for each piece by name and tell them where to place them, how much to rotate each one and which direction to rotate each piece – clockwise or counterclockwise. I would probably also have to give details about how to align each one relative to the other pieces or relative to the state. I also might refer to the piece regarding their relationship to the top or sides of the work area or template on which I'm asking them to be arranged. Since each piece has a certain number of corners, I would refer to each corner point on each piece and explain how it might fit relative to the corner points on other pieces. I'd probably refer to each point as topmost, leftmost, bottommost or right most. I might have to tell them to nudge each piece up, down, left, right, or a combination of directions.

If I was telling a computer how to render and arrange the pieces using actionscript I would probably need to use the following classes, methods and parameters.

First I'd probably use the MovieClip Object, which is an extension of the DisplayObject Class. Here's the inheritance structure:

```
MovieClip < Sprite < DisplayObjectContainer < InteractiveObject < DisplayObject <
EventDispatcher < Object
```

Each piece could be drawn and stored as a movie clip. I'd start by creating a movieclip for each piece and storing that within an array:

```
var myMovieclip0:MovieClip = new MovieClip();
var myMovieclip1:MovieClip = new MovieClip();
var myMovieclip2:MovieClip = new MovieClip();
var myArray:Array = new Array();
myArray[0] = myMovieClip0;
myArray[1] = myMovieClip1;
myArray[2] = myMovieClip2;
```

You could also use a for loop to assign each movie clip to an array.

```
var i = 0
for (i=0; i< numberOfPieces; i++){
    myArray[i] = myMovieClip[i];
}
```

MovieClips have lots of helpful parameters that are inherited from the display object class that would help me place each item on the stage in the right position and rotate each object as needed. These parameters would be helpful. I would refer to some display object parameter through the movie clip extension by name and use the dot syntax to refer to the following parameters.

To change x position:
myArray[0].x
To change y position:
myArray[0].y
To change rotation:
myArray[0].rotation

Inside each movie clip I would draw the shape of each piece by defining where each point is then drawing between the points. I could use the Point Object to create and store the points x and y values, basically considering it on a grid where top left is zero. I would define each point and then draw between each point like below:

```
var point1:Point = new Point();  
trace(point1); // (x=0, y=0)  
var point2:Point = new Point(50,50);  
trace(point2); // (x=50, y=50)  
and so on....  
var trapezoid:Shape = new Shape();  
trapezoid.graphics.lineStyle(10, 0xFFD700, 1, false,  
trapezoid.graphics.moveTo(point1.x, point1.y);  
trapezoid.graphics.lineTo(point2.x, point2.y);
```

and so forth...

When I move the shapes I could reset the position of each point and store it globally so that the pieces could move relative to each other. The rest would involve using variables and conditionals to tell the position of the pieces and their points relative to each other and the stage. Many functions could be developed to control animation more articulately, but these are the basics objects and parameters needed.