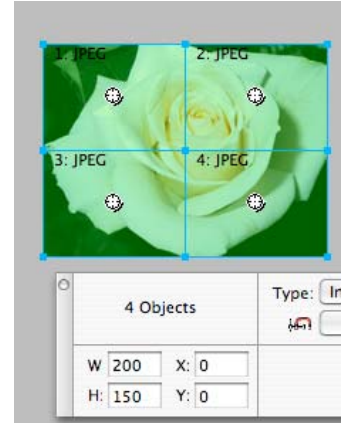


# Flash Jigsaw Puzzle

The first decision is the actual photograph/drawing (know as photo throughout this tutorial) that will be used for the puzzle. For the first puzzle use something simple that can be divided into 4 pieces evenly.

## *Slicing up the photo in Fireworks*

1. Open the photo in Fireworks. [Reminder: Photos should be JPG; Drawing without any gradient should be GIF. Optimize this in Fireworks if it is not either.]
2. Modify > Image Size and reduce the photo to a reasonable width. The example display here was reduced to 200 pixels wide.
3. Drag guidelines from the ruler to divide the photo into equal pieces. Guidelines just help visualize the puzzle pieces for the next step.
4. Using the Slice tool, draw one slice over one section of the guideline-divided photo. The Properties panel also helps to change the width and height of that slice to be accurate.
5. Draw one slice over every other section.
6. Name each section in consecutive order. This helps in exporting from Fireworks and importing into Flash.
7. File > Export. Select "Image Only" for the Save As option in this dialog box. Export to a known folder since it will export each slice to its own jpg file.
8. One complete photo is also needed. The simplest without creating a whole new file is File > Export. Select "None" under Slices. The whole photo will not export as a jpg without the slices.
9. Then save the original file as a PNG just in case it needs re-slicing.



## *Starting the Flash puzzle*

1. Open Flash.
2. Size the movie to fit the puzzle with room for the puzzle pieces mixed up on one side and the puzzle put together on the other side.
3. In the Timeline, create seven more layers. Name the layers from top to bottom: actions, labels, counter, pieces, places, text, playAgain, puzComplete
4. File > Import > Import to Library the puzzle pieces and the whole photo exported from Fireworks.

There are going to be many movie clips and actionscripts for the places and pieces. Each layer will have its own set of instructions. Make sure to select before working on that layer!

## *actions Layer*

1. Put keyframes in the Timeline at frame 2 and 3. [F6 is the shortcut key for inserting a keyframe.]
2. Insert a frame at frame 4. [F5 is the shortcut key for inserting a frame.]
3. On frame 1, insert action

```
hitCount = 0
```

This actionscript creates the variable to keep count of pieces when placed.

4. On frame 2, insert action
- ```
hitCountText = hitCount  
if (hitCount == 4) {  
    gotoAndStop ("end");  
}
```

This actionscript checks the text box for a visual text of the hitCount variable then if hitCountText equals the amount of the hitCount and all pieces are placed then goes to "end" frame.

5. On frame 3, insert action
- ```
gotoAndPlay("loop");
```

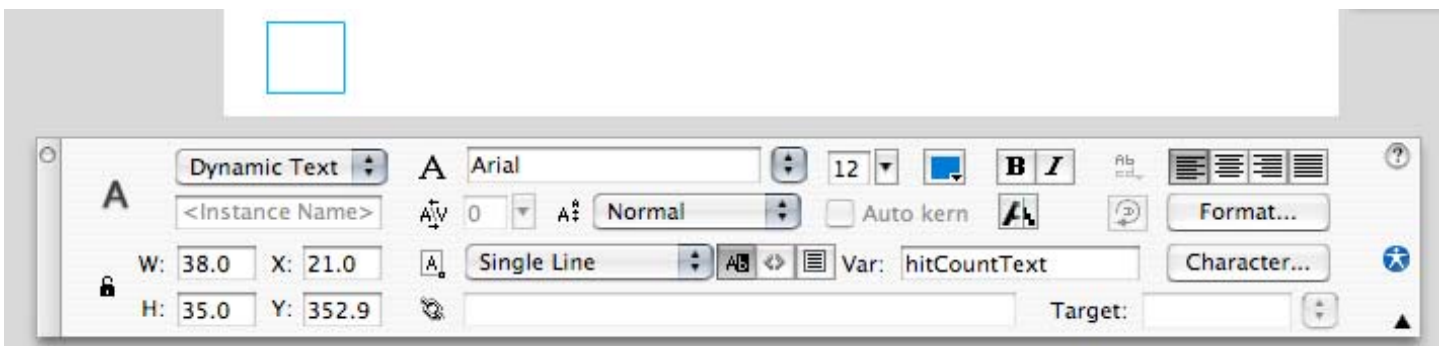
This actionscript returns to loop frame and check hitCount.

### **labels Layer**

1. Put a keyframe in the Timeline at frame 2 and 4.
2. Enter a frame label these keyframe:  
Frame One: start  
Frame Two: loop  
Frame Four: end

### **counter Layer**

1. Create a dynamic text box in one of the lower corners of the stage. [If you don't want the counter visible while playing the puzzle, you can place the counter off-stage. Or leave it on the stage until you finish creating the whole puzzle then move it off-stage. It is a great indicator that all puzzle pieces have been placed and the puzzle is working.]
2. Format the dynamic text box with the font style, size and color. This can always be changed.
3. Also in the Properties panel, name the variable to:  
`hitCountText`



4. At frame 3 insert a frame.

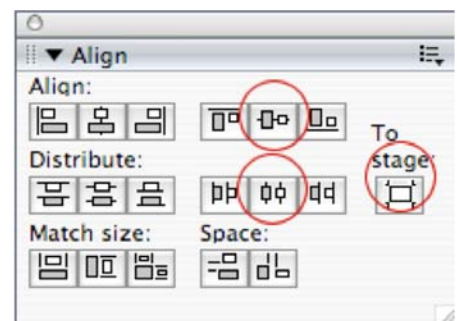
### **pieces Layer**

Here's where it gets complicated!

1. Drag each jpg from the library to the right side of the stage. Try to keep them in some order, but not necessary to have them placed just like the puzzle.
2. Start with puzzle piece 1. Select its jpg and Convert to Symbol either by right-clicking and selecting Convert to Symbol or the menu item Modify > Convert to Symbol.
3. Name this symbol `pieceButton1` and make sure it has the behavior of a button.
4. Now immediately Convert to Symbol again, naming it `pieceMov1` and make sure it has the behavior of a movie clip.
5. Name this symbol instance:  
`piece1`
6. Double-click this symbol to edit it.
7. Name the existing layer  
`Button`

Double-click the button. Using the align panel, select "To Stage". Then click center horizontally and center vertically (see diagram). Having your puzzle piece centered on the symbol's center registration mark, your puzzle piece will release from the mouse on stop drag in step 9.

8. Add a new layer and name it  
`actions`  
Add this actionscript to frame 1 of actions layer  
`stop();`  
Add a keyframe in frame 2 of the actions layer.



9. Selection the button in frame 1 of the button layer. Add this actionscript to that button:

```
on (press) {
    startDrag(_parent.piece1, true);
    mx.behaviors.DepthControl.bringToFront(_parent.piece1);
}

on (release) {
    stopDrag();
    if(_parent.place1.hitTest(_root._xmouse, _root._ymouse)) {
        _parent.hitCount = Number(_parent.hitCount)+1;
        _parent.place1._alpha = 100;
        gotoAndStop(2);
    }
}
```

Explanation of the code:

The first section tells the movie that if the mouse is held down, this button will drag the movie clip and also bring that movie clip to the front.

The second section tells the movie to stop dragging the movie clip, then check to see if the mouse is over the corresponding place movie clip. If yes, increase hitCount by 1, change place movie clip transparency, and go to blank frame in piece movie clip.

10. Return to Scene 1. Select piece1 and add this actionscript

```
onClipEvent(load) {
    _x = Math.floor(Math.random() * (560 - 400) + 400);
    _y = Math.floor(Math.random() * (400 - 80) + 80);
}
```

The random placement of this movie clips is controlled by the figures within the boundaries of the right side of the screen in this sample. `_x` is the horizontal and `_y` is the vertical. The actionscript will vary according to the size of the stage and the size of the movie.

11. For each of the remaining puzzle pieces, follow steps 2-9, just changing the piece number and place number in Step 8 actionscript and Instance Name in Step 5 to correspond with the puzzle piece's number.
12. Add a frame in Frame 3 of pieces layer.

### ***places Layer***

1. Drag each jpg from the library. Place them just like the puzzle.
2. Start with puzzle place 1. Select its jpg and Convert to Symbol either by right-clicking and selecting Convert to Symbol or the menu item Modify > Convert to Symbol.
3. Name this symbol place1 and make sure it has the behavior of a movie clip.
4. Name this symbol instance:

```
place1
5. Add this actionscript to this symbol
onClipEvent(load) {
    this._alpha = 10
}
```

This actionscript controls the movie clip's visibility. The "10" can be changed to whatever figure you would like.

6. For each of the remaining puzzle places, follow steps 2-5, just changing the piece number and Instance Name in Step 4 to correspond with the puzzle place's number.
7. Add a frame in Frame 3 of places layer.

### ***text Layer***

1. Add a keyframe in Frame 4 of this layer.
2. With the text tool, draw a static text box. Type  
    Congratulations!  
in that text box.

3. Move the box to either the right or bottom of the stage. Its placement can change after you do the next two layers.

### ***playAgain Layer***

1. Add a keyframe in Frame 4 of this layer.
2. With the text tool, draw a static text box. Type  
Play Again?  
in the text box.
3. Convert this text box to a button symbol call playAgain. Using your knowledge in creating buttons in Flash, edit the button to have a rollover and a hit rectangle.
4. Return to Scene 1 and add this actionscript to the button on the stage

```
on (press) {
    gotoAndPlay("start");
}
```

### ***puzComplete Layer***

1. Add a keyframe in Frame 5 of this layer.
2. Drag the whole photo jpg to the stage. Position it where you would like.

TEST YOUR MOVIE! Now you can export it for insertion into a webpage.