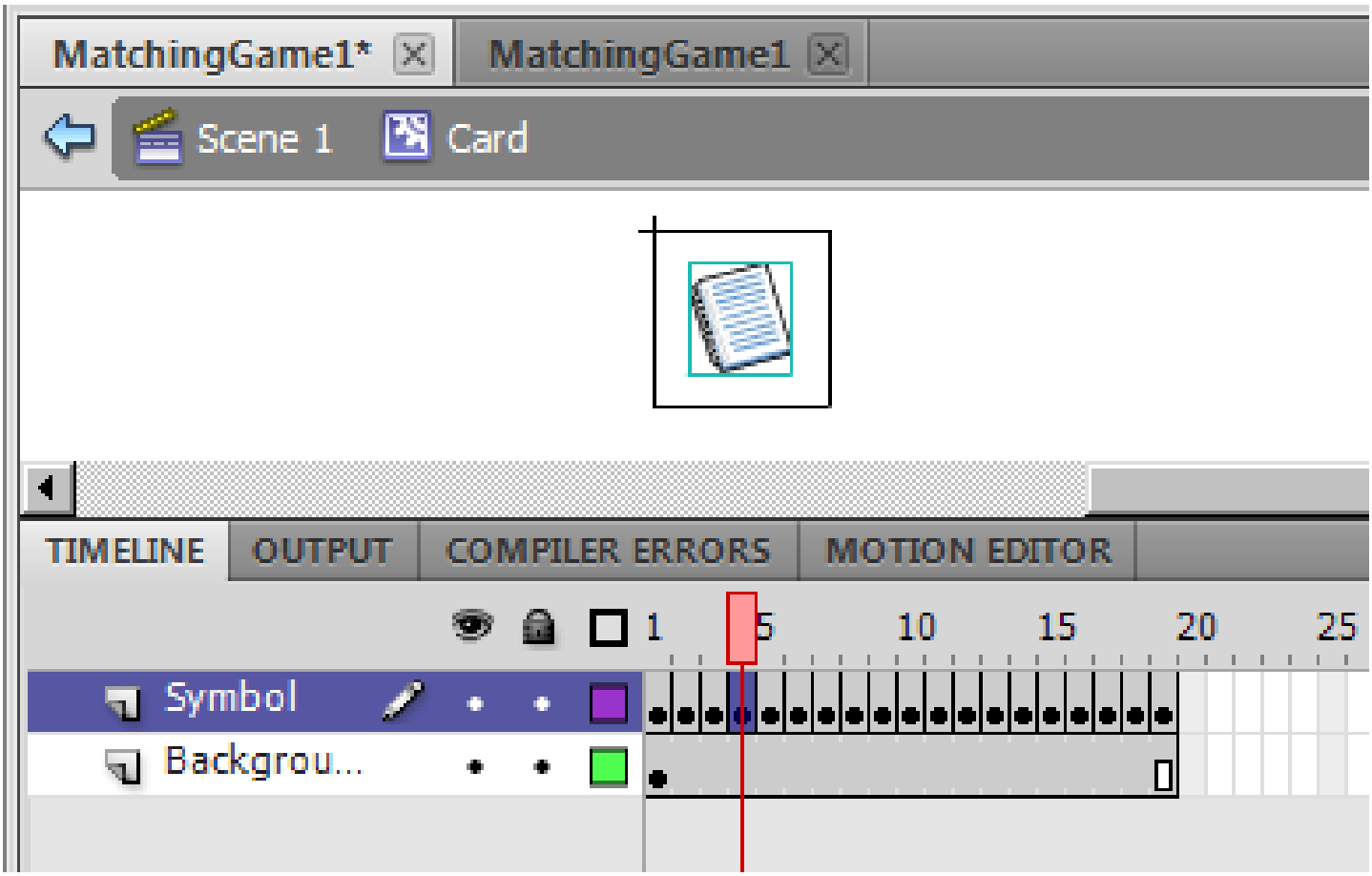
1. Open up the `MatchingGame1.fla` and the `MatchingGame1.as` files. You may wish to make copies of them so you still have the original unchanged version of each.
2. Try out the `MatchingGame1.fla` application. Hold down the Control key and press Enter key to Test Movie using the shortcut.
3. Comment out the `thisCard.stop();` statement using two forward slashes. **`// thisCard.stop();`** After saving the MatchingGame1.as ActionScript 3.0 file, run the application again, i.e. Test Movie.
4. Now add the following statement to the code, while still leaving thisCard.stop(); statement commented out with // as it is: **`thisCard.gotoAndStop(Math.floor(Math.random() * 19 + 1));`** Then save your work and run the program again.

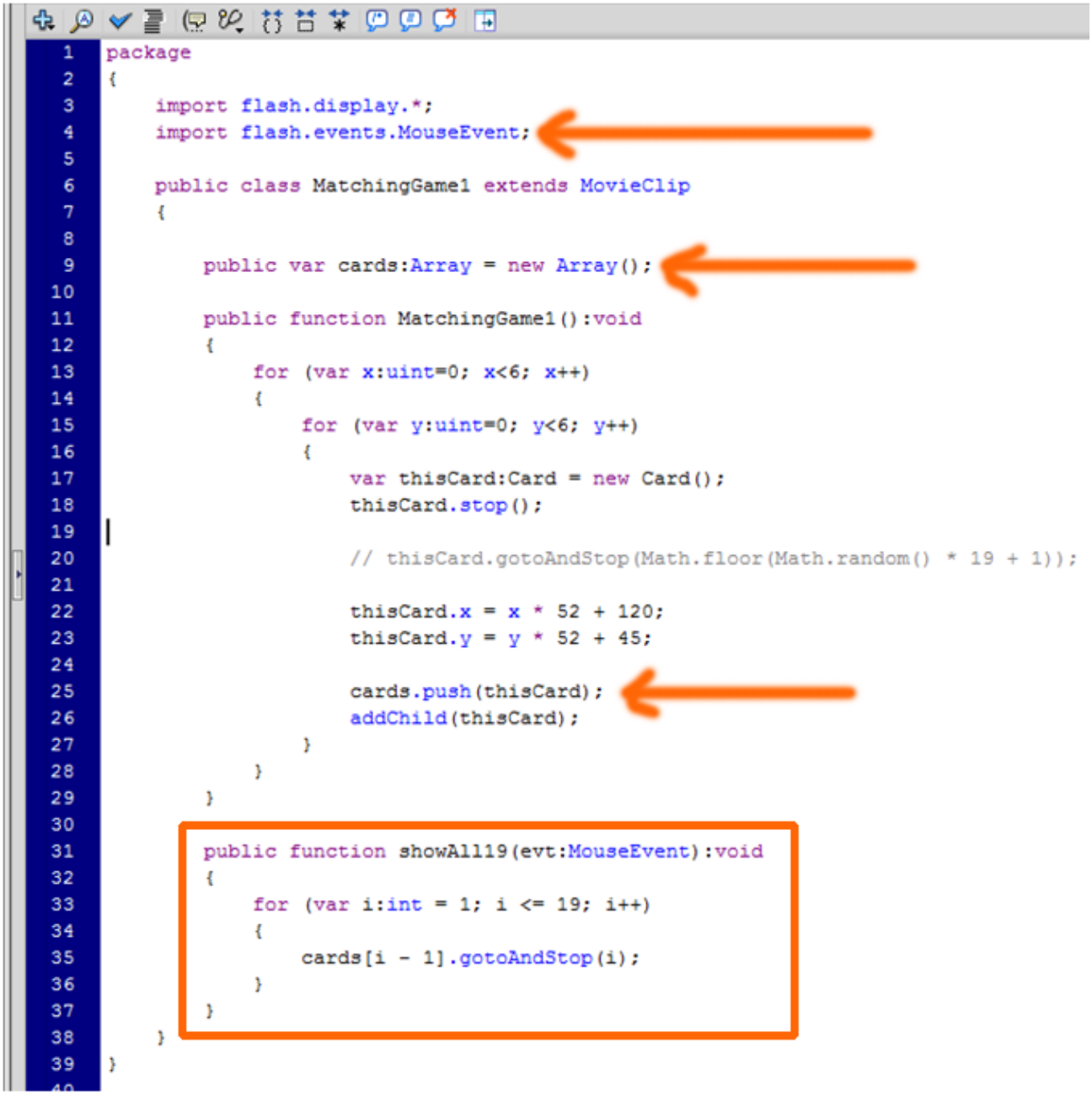


5. Recall that Control+L is the shortcut to see the Flash Library. There is only one Symbol for this application. It is named Card and is type Movie Clip. Here is what the Card symbol looks like. It has 19 different frames. Click the eyeball column switch for the Background layer to see what the background graphic is. What is Frame #2? What is Frame #10? What is Frame #12? What is Frame #16? What is Frame #18? What is Frame #1?
6. Run your application again. Did you get at least one of each of the graphics you wrote down? (Frame #1, #2, #10, #12, #16, and #18).
7. Add a button to your user interface. Name the button **`showAll19_btn`** for its instance name. Add the following code to the application, in a separate layer named **`Actions`**:

```
showAll19_btn.addEventListener(MouseEvent.CLICK, showAll19);
```

8. Now go back to the ActionScript 3.0 MatchingGame1.as file. Add the following 10 lines of code. Also, change the commenting (the // comments) as shown here so the thisCard.stop() is again in effect and the

`thisCard.gotoAndStop(Math.floor(Math.random() * 19 + 1));` is // commented out.



```
1  package
2  {
3      import flash.display.*;
4      import flash.events.MouseEvent;
5
6      public class MatchingGame1 extends MovieClip
7      {
8
9          public var cards:Array = new Array();
10
11         public function MatchingGame1():void
12         {
13             for (var x:uint=0; x<6; x++)
14             {
15                 for (var y:uint=0; y<6; y++)
16                 {
17                     var thisCard:Card = new Card();
18                     thisCard.stop();
19
20                     // thisCard.gotoAndStop(Math.floor(Math.random() * 19 + 1));
21
22                     thisCard.x = x * 52 + 120;
23                     thisCard.y = y * 52 + 45;
24
25                     cards.push(thisCard);
26                     addChild(thisCard);
27                 }
28             }
29         }
30
31         public function showAll19(evt:MouseEvent):void
32         {
33             for (var i:int = 1; i <= 19; i++)
34             {
35                 cards[i - 1].gotoAndStop(i);
36             }
37         }
38     }
39 }
```

Line 4 is needed for the public function showAll19. Line 9 declares the Array for holding and remembering the 36 cards. This array is needed so we can go back to them and refer to them later with code. Specifically, line 35 will be executed 19 times, on 19 of the 36 different cards.

Lines 31 to 37 are new code for defining the showAll19() function. Why does it need to be a PUBLIC function?

Line 25 is cards.push(thisCard); It adds the given card to the array which we have given the name cards when we defined it on line 9.

Line 35 uses the cards array. cards[i – 1] refers to cards[0] when i = 1, refers to cards[1] when i = 2.

9. Experiment: Go back to the code now and try making the following change: Make the 1st for loop use the x variable and make the 2nd for loop use the y variable. In other words, switch x and y around. Line 13 is the y loop, and line 15 is the x loop. Rerun it and click the button. What is the difference in the output of the showAll19_btn click results?

```
13          for (var x:uint=0; x<6; x++)
14          {
15              for (var y:uint=0; y<6; y++)
```

10. What is the MOD function in ActionScript 3.0 language? The % symbol specifies the MOD function. The % operator gives you the remainder of the division. Dividend % Divisor is the format for the operation. Modify the showAll19 function as shown here:

```
public function showAll19(evt:MouseEvent):void
{
    for (var i:int = 0; i < 36; i++)
    {
        var r:int;
        r = i % 19;
        cards[i].gotoAndStop(r + 1);
    }
}
```

What is the remainder when you divide 0 or 19 by 19? The remainder is 0. Only the quotient is different.What is the remainder when you divide 1 or 20 by 19? The remainder is 1. Only the quotient is different.

11. Rerun your application and click the button to see all 36 cards. Which cards do NOT get repeated at least once?

12. Finally, add a new button named **show18_btn** to your .fla Flash application. When you get done, your actions layer will consist of the following two statements:

```
showAll19_btn.addEventListener(MouseEvent.CLICK, showAll19);

show18_btn.addEventListener(MouseEvent.CLICK, show18);
```

13. The .as ActionScript 3.0 Class file for the MatchingGame1 class should now be modified to include the public function show18(), which is shown to the right here:

Note that when whichCard reaches 20, it gets set back to 2 again.

```
public function show18(evt:MouseEvent):void
{
    var whichCard:int = 2;
    for (var i:int = 0; i < 36; i++)
    {
        cards[i].gotoAndStop(whichCard);

        whichCard++;
        if (whichCard > 19)
        {
            whichCard = 2;
        }
    }
}
```