

ActionScripts II: Firecracker

Topic Covered: Dragging a symbol on the stage, If Then statements

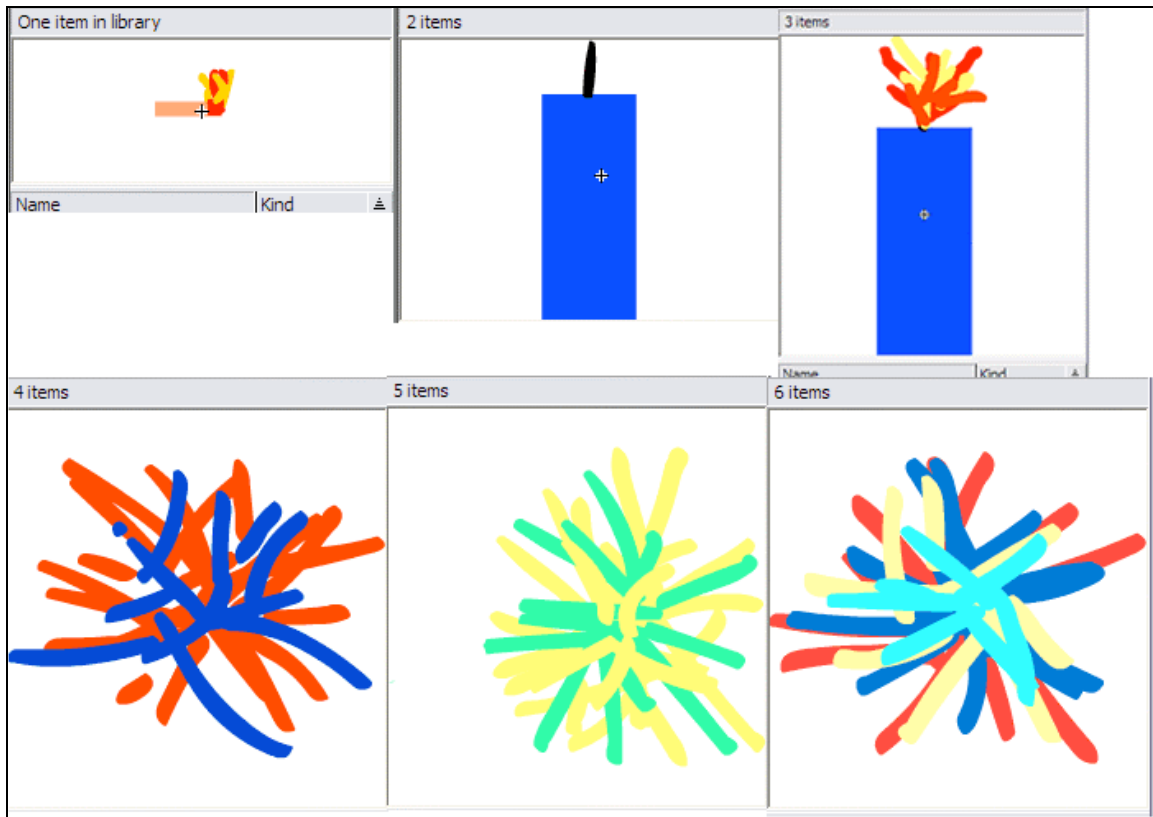
In this lesson you will learn how to use ActionScripts to make an object move with a mouse and make objects on the screen interact. You will create a match and a firecracker. The firecracker will explode when you touch it with the match. The explosion is created by creating single frames of scattered color.

Let's get started. First, you need to create the library items you will need:

- “match” – a movie clip
- “firecracker” – a movie clip
- “firecracker-lit” – a graphic or a movie clip

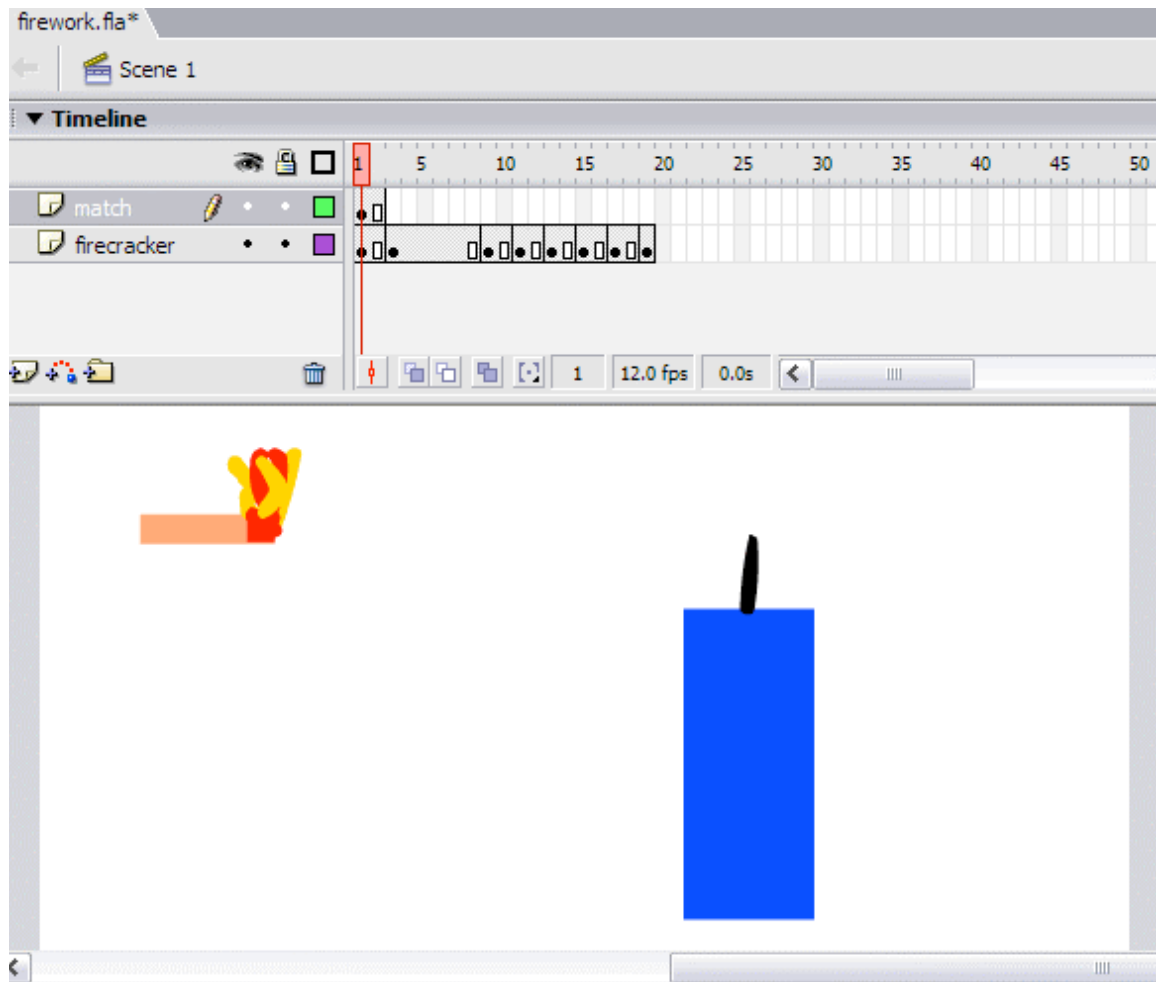
and three symbols with scattered color for the explosion itself—

- “explosion1” – a graphic
- “explosion2” – a graphic
- “explosion3” – a graphic



Next, you need to create a 20 frame timeline. Most of the timeline will be used for the explosion. You are going to use ActionScripts to stop and start the timeline. Put “match”

and “firecracker” in frames 1 and 2 on separate layers. Put “firecracker-lit” in frames 3-9 where “firecracker” was (onion skinning anyone). Put “explosion1” in frames 11-12 where the firework used to be, “explosion2” in frames 13-14, “explosion3” in frames 15-16, and then repeat the explosions for 2 frames each until frame 20. You will need keyframes at frame 1, 2, 10, and then every other frame until frame 20. Your timeline should look like this:



Add a layer for you ActionScripts (remember to call it Actions) and put a keyframe in frame 1 and a keyframe in frame 2.

Rather than have the movie stop, what you want it to do is move in a small loop between frames 1 and 2 because you want it to continue checking to see if the match has touched the firecracker. Let’s set the loop up first. In frame 2 of the Actions layer, type in “gotoAndPlay (1);” This will automatically send the movie back to frame 1. Next, let’s make the match follow the cursor once you click the match. Some of this will be familiar from ActionScripts I. You need some instance names to use in the ActionScript. Click on the match on the stage. Change the instance type in properties to Movie Clip if it is not

already set that way and set the instance name to “burn.” This will allow you to reference the match in your ActionScript. In the Actions layer on frame 1, enter:

```
burn.onRelease = function () {  
    startDrag(burn, true) }  
}
```

What you’ve done is told Flash that when the match has been clicked, it should follow the cursor from then on.

Try previewing the file. You should be able to move the match once you have clicked on it, but you should not be able to see anything that happens past frame 2—let’s solve that.

The next thing you need to do is write a script so that when the match touches the firecracker, the movie will continue and the firecracker will appear to explode.

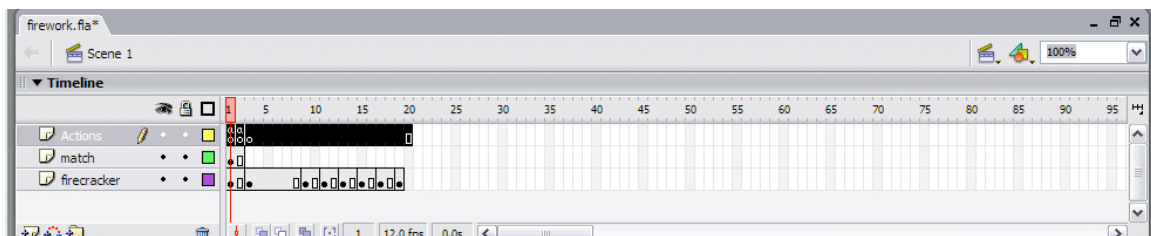
As with the match, you need to make the firecracker a movie clip if it is not already set that way. Set its instance name to “lightme.” Open the action script in frame 2 again and modify it so it looks like this:

```
if (burn.hitTest(lightme)){  
    gotoandplay (3);  
}  
else  
    gotoandplay(1);
```

What you’ve done is told Flash to see if the match (burn) has intersected with the firecracker (lightme). If it does, then Flash goes to frame 3, otherwise (else) it goes back to frame 1.

Try previewing the file.

Here’s what the final timeline should look like:



Firecracker 2.0

Here's another way to move the match.

Delete the ActionScript in Frame 1 and add the following ActionScript to the match instance (burn).

When you create the match, make it a movie clip. Put it on its own layer. With the mouse selected, apply the following ActionScript

```
on (press) {  
    this.startDrag();  
}  
on (release) {  
    this.stopDrag();  
}
```

This allows you to move the match while you have the mouse button down. Make sure you use the check script feature after you enter the script—if you see an error, you probably put the script in frame 1 rather than on the “burn.”

If you want it to move when you release the mouse button, use this script:

```
on (release) {  
    this.startDrag();  
}
```

You could set a variable to check if the user was already dragging the mouse and stop the dragging the second time the user released the mouse button.

Taking things further:

1. Get the flames on the match to flicker.
2. Get the match to light when you are moving it and stay unlit when you are not moving it.
3. Get the firecracker to **only** explode when you touch the fuse, not the another part of the firecracker
4. Do the same with the head of the match.