

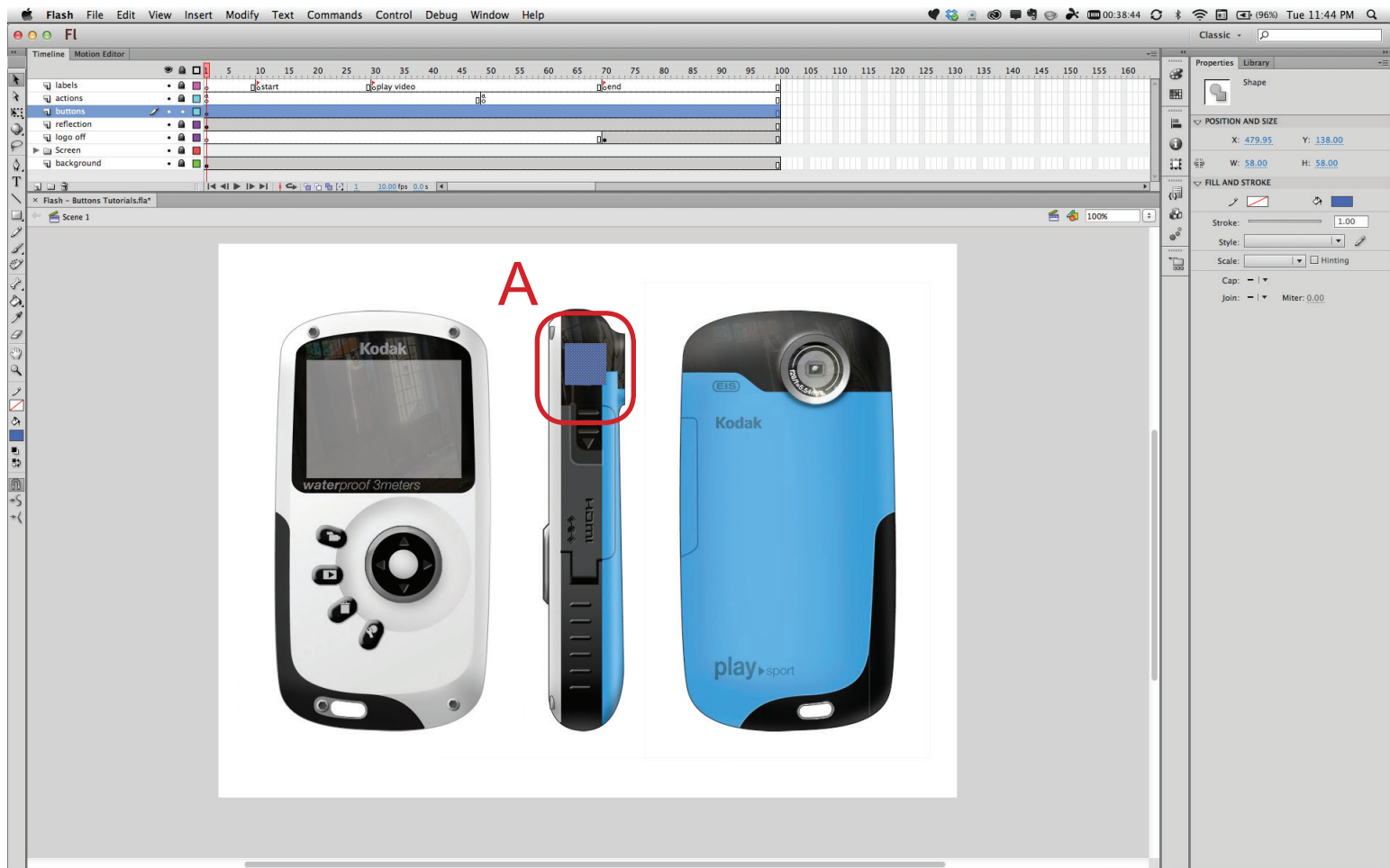
Flash - Buttons & Navigation

This tutorial will cover how to setup a Flash file with a button MovieClip (A) and apply coding to make the it control the navigation through the Flash file timeline or timeline of MovieClip (B).



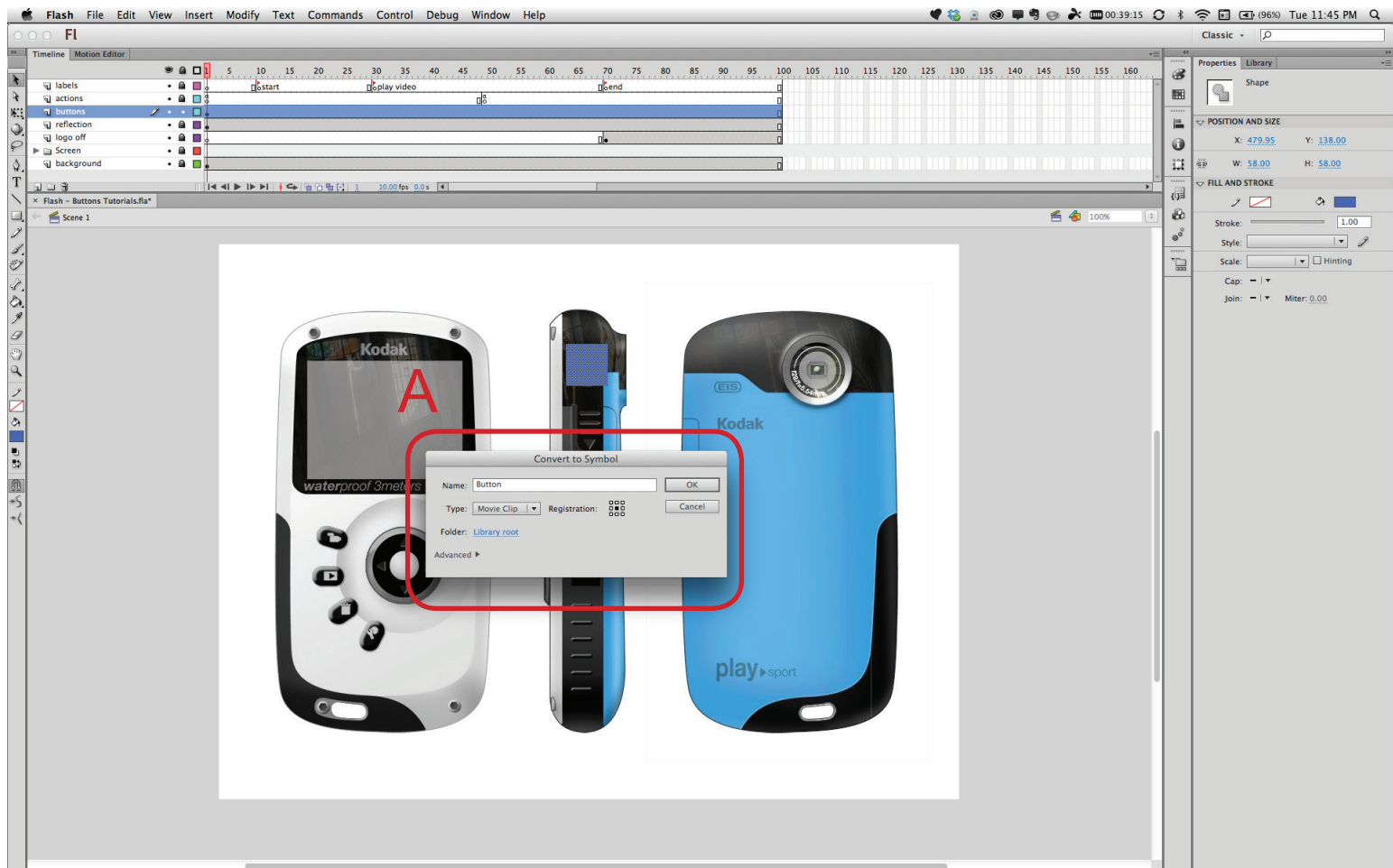
Flash - Buttons & Navigation

Step 1 To create the button element, draw a graphical shape (fill only, no stroke) over the area for the button (A).



Flash - Buttons & Navigation

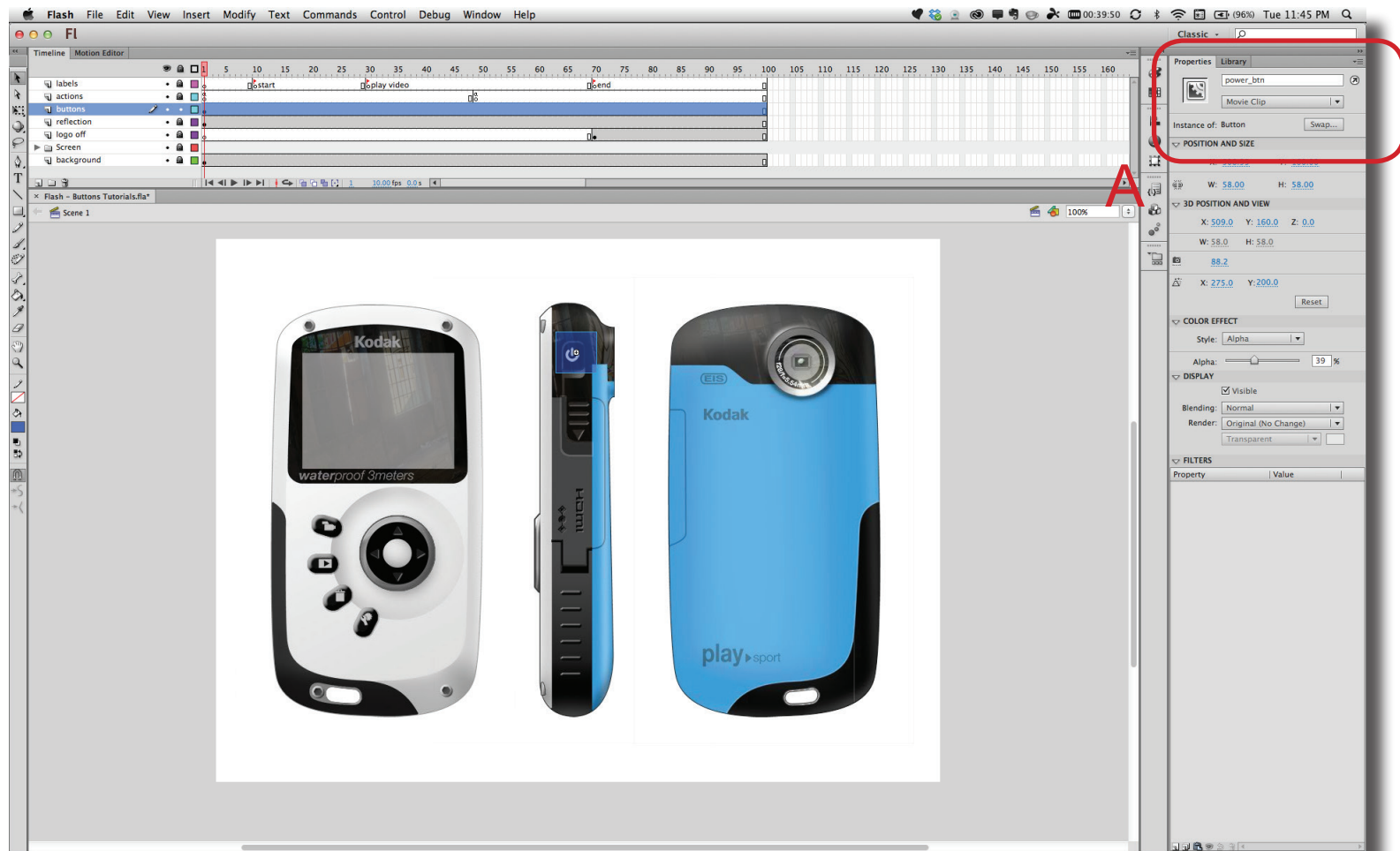
Step 2 Next, convert the element to a MovieClip by selecting the graphics on the stage and going to Modify > Convert to Symbol. In the dialogue box, Name the symbol, verify Type is MovieClip and set the Registration point to the center (A).



Flash - Buttons & Navigation

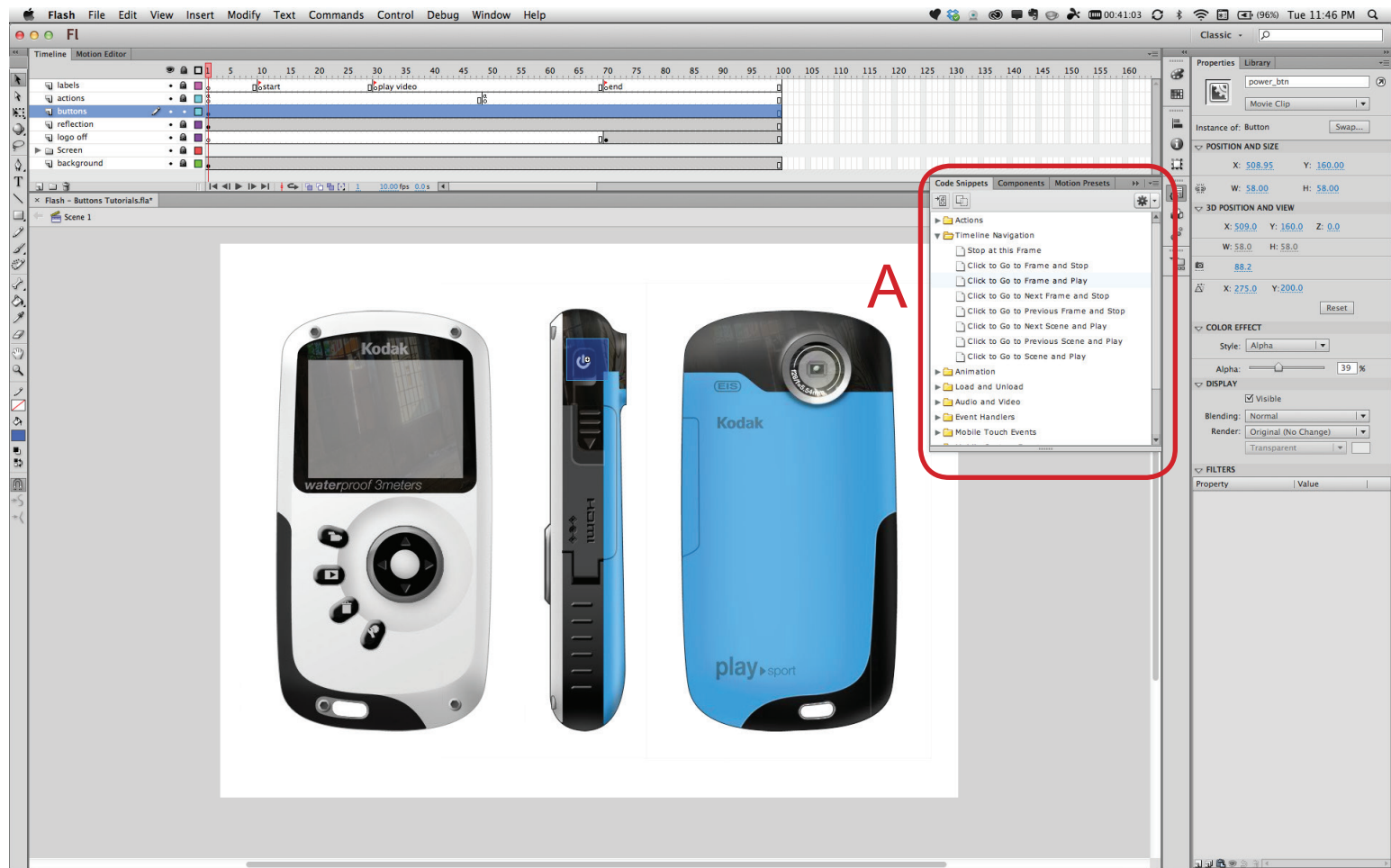
Step 3

To reference the button graphics with code, the instance of the library Button on the stage needs to be named. Select the button element on the stage and enter a name in the Instance Name block under the Properties panel (A). Instance names cannot have spaces, start with a number and must be unique. In this example, “power_btn” is being used.



Flash - Buttons & Navigation

Step 4 The Code Snippets panel can help with writing the ActionScript code required to make the button element active. Make sure the button element is selected. Then, go to Window > Code Snippets and open the Timeline Navigation folder (A). To add the code to the Timeline, double click on the “Click to Go to Frame and Play”. If an “Actions” layer is not already in the Timeline, then this layer is added and the code is placed in the first frame.



Flash - Buttons & Navigation

Step 5 The code that is added is shown in the Actions panel which is open automatically (A). The code is generic and needs to be customized. Two elements of code were added; one relative to the button element (listener) (B) and second relative to the button's action (function) (C).

B power_btn.addEventListener(MouseEvent.CLICK, fl_ClickToGoToAndPlayFromFrame);

Instance Name Listener Event Function

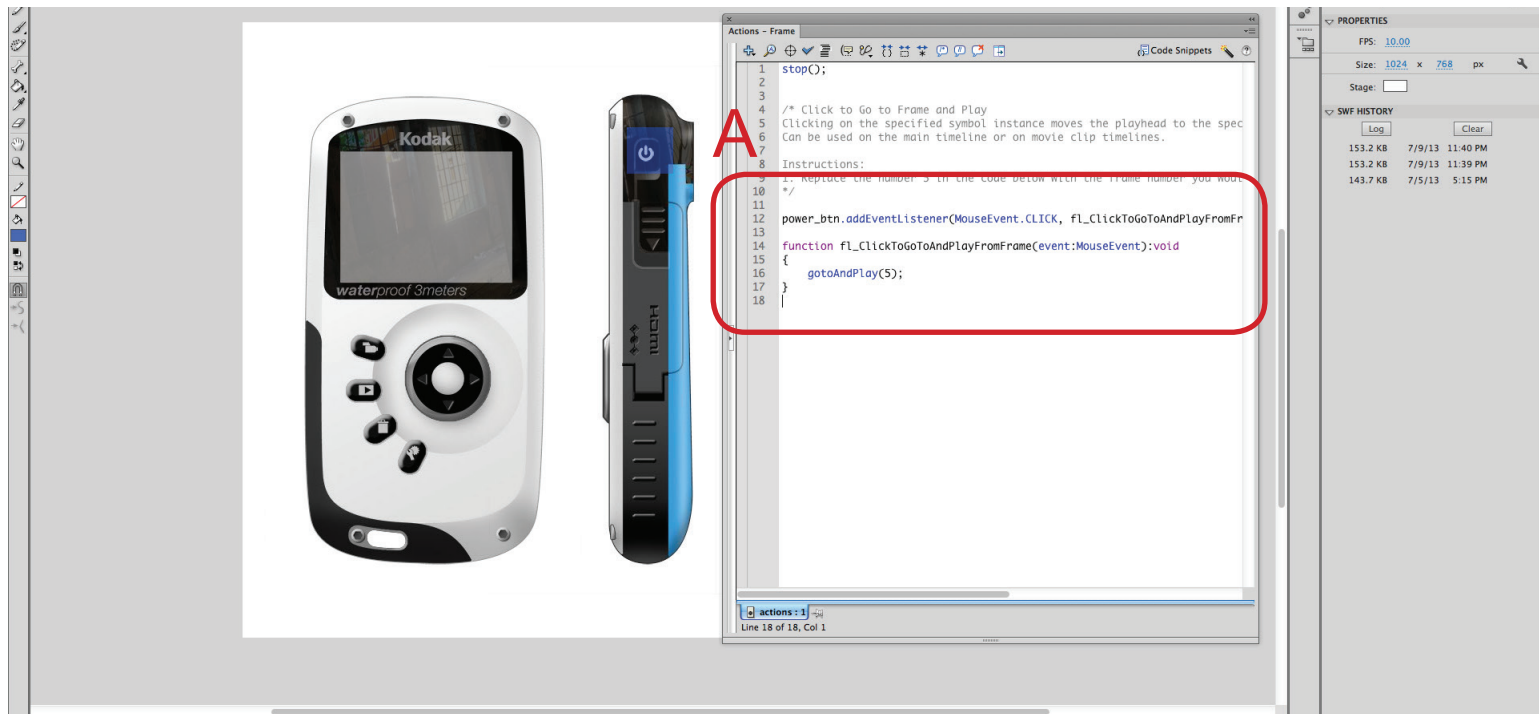
C function fl_ClickToGoToAndPlayFromFrame(event:MouseEvent):void

{

 gotoAndPlay(5);

}

Function Event Custom Code



Flash - Buttons & Navigation

Step 6

To customize this text, the function name needs to be changed and the code within the function needs to be set. Any name for the function can be used but for convenience, the instance name of the button element will be reversed (“power_btn” -> “btn_power”) and replaced in two locations (A) and (B).

power_btn.addEventListener(MouseEvent.CLICK, btn_power); A

Instance Name Listener Event Function

B

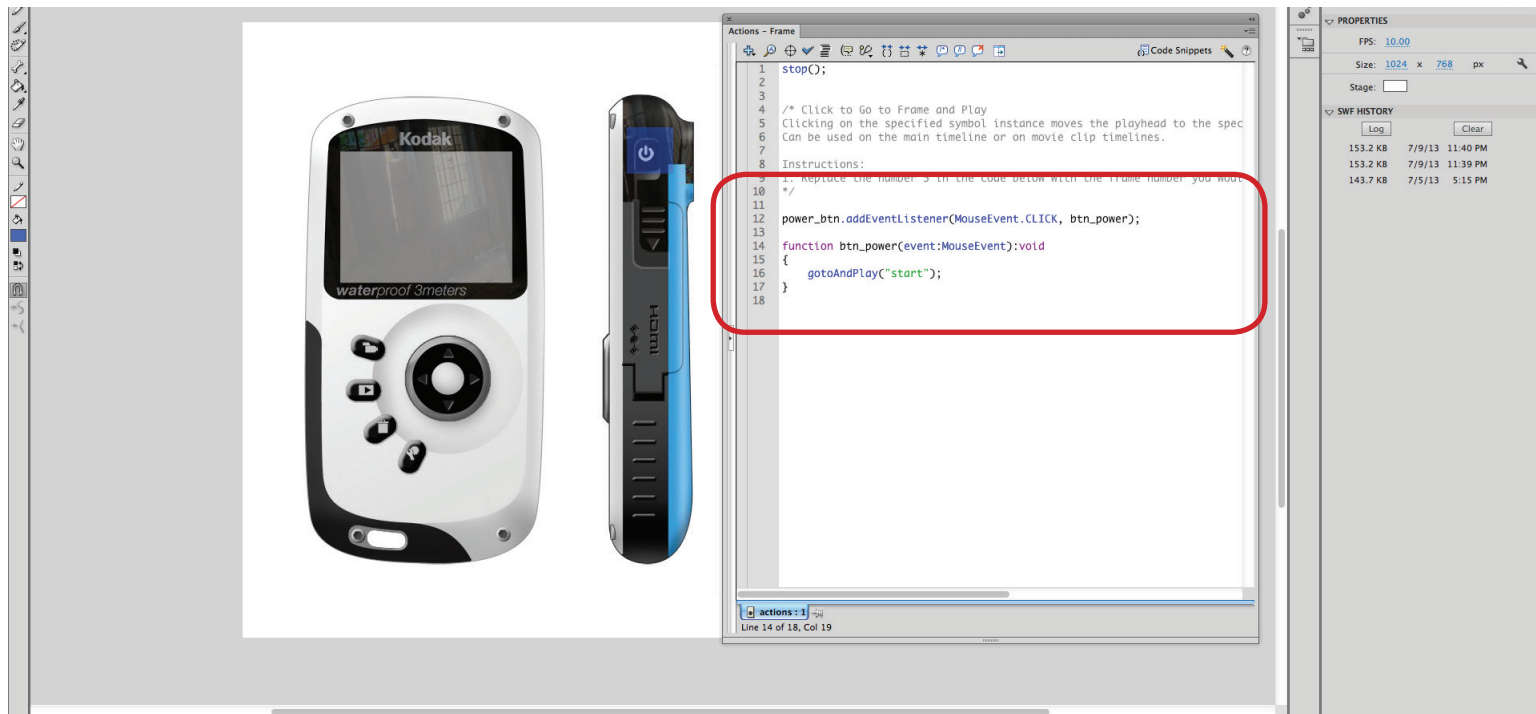
function btn_power(event:MouseEvent):void

Function Event

{

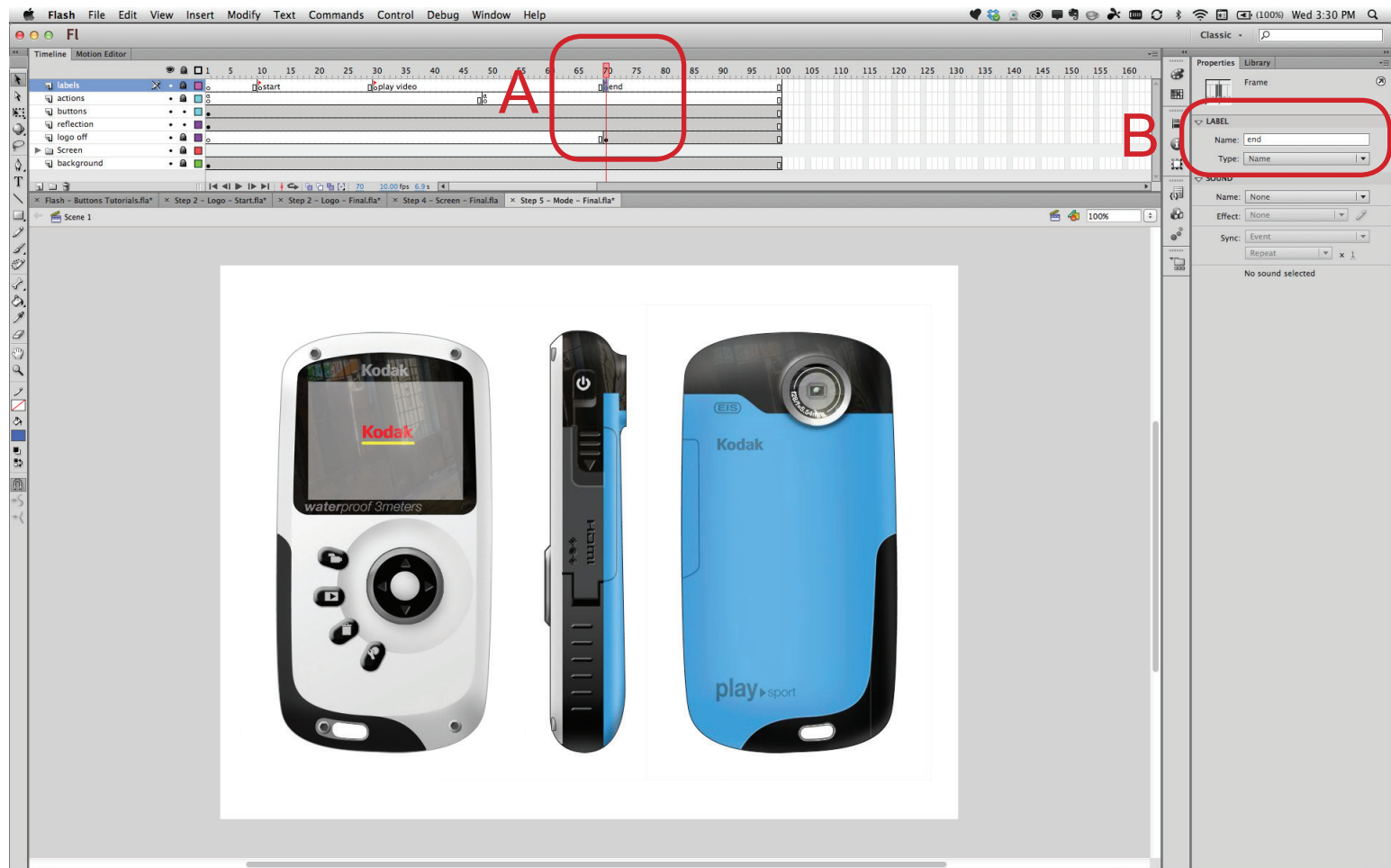
gotoAndPlay(5); Custom Code

}



Flash - Buttons & Navigation

Step 7 To navigate in the Timeline without having to worry about specific frame numbers, frame labels can be added to a specific frame in the Timeline. If frames are added or removed from the Timeline, then labels will shift around appropriately and can still be referenced. Select the frame (usually at a Keyframe) (A) and in the Property panel a name in the Label > Name section (B).



Flash - Buttons & Navigation

Step 8

Below is the code that was added to the “btn_power” function. Two types of timeline navigation are shown. First, an if/else statement is used to evaluate the location of the main time (currentFrame). If the frame number is equal to 49 (true), then the main timeline jumps to the frame label “end” and starts playing (A). The second timeline navigation is for the “logo_off” MovieClip. A dot notation (“.”) is used to tell the timeline in this MovieClip to start playing at frame 1 (B).

```
if (currentFrame == 49) {  
    gotoAndPlay("end");  
    logo_off.gotoAndPlay(1);  
} else {  
    gotoAndPlay("start");  
    logo_on.gotoAndPlay(2);  
}
```

