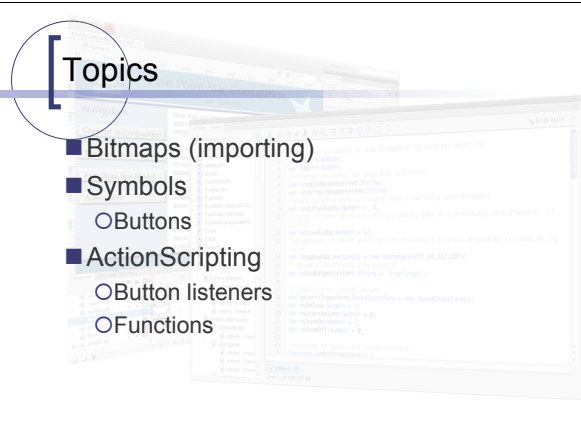


Button Symbols, Bitmaps, Actionscripting

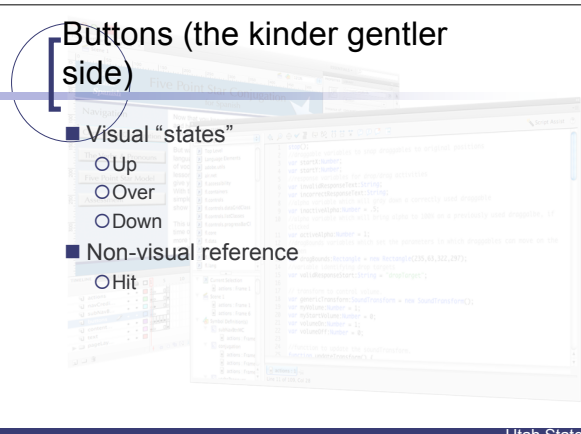
Utah State



Topics

- Bitmaps (importing)
- Symbols
 - Buttons
- ActionScripting
 - Button listeners
 - Functions

Utah State



Buttons (the kinder gentler side)

- Visual "states"
 - Up
 - Over
 - Down
- Non-visual reference
 - Hit

Utah State

Buttons (Actionscript side)

- Can be told to listen for mouse events (rollover, click, release, etc . . .)
- ActionScript is **separate**

Utah State

Button action scripting in CS3/CS4

- (assuming ActionScript 3.0)
 - The five have become one.
 - DOM inspired event handlers
 - Custom Functions are no longer optional.

Utah State

Function – basic concept

- Do my work for me (think automatic response)
 - Responding to a greeting
 - Accept Input (e.g. "hey how's it going?")
 - Process (determine appropriate response)
 - Output response (e.g. "good, how are you?")
 - The function does the work
 - Doesn't handle unanticipated inputs well (e.g. "high five")

Utah State

Using Functions

- Call function name
- Add parameters, e.g. inputs (if necessary/desired) in parentheses
- Parentheses **not** optional
- End in a semi-colon
- Examples:

```
stop();  
gotoAndStop(2);  
addEventListener(MouseEvent.CLICK, onWindowDocBtnClicked);
```

Utah State

Creating functions

```
function functionName (inputName:inputType) {  
    //code to run when function is called  
}
```

Example:

```
function clickMyButton(e:MouseEvent) {  
    gotoAndStop(3);  
}
```

Utah State

Instance names

- Actionscript reference to a symbols **instance**
- Different than a symbol name
- Different namespace than symbol names
- Assign using the properties window
- Or Target Path

Utah State

Functions that will help in assignment 6

- `stop();`
- `nextFrame();` // takes you to the next frame in the timeline.
- `prevFrame();` // takes you to the previous frame in the timeline.
- `gotoAndStop(1);` // takes you to the first frame and stops the timeline. (can replace 1 with any number you want).
