

Movie Clips, Dot Syntax, Dynamic Text



Utah State

Topics

- Movie Clips
- Repetitive Animation “loops”
- Using actionScript to control “other” timelines
 - Instance names
 - `root`, `parent` (object hierarchy and dot syntax)
 - Variable Casting
- Dynamic Text
- Combining Motion

Utah State

Movie clips

- Uber symbol
- **Independent** timelines
- Mini flash files

Utah State

Combining Motion

- Car Driving off screen (example)
 - Looping and combining
- Ball revisited (example)
 - Combining
 - copy|paste frames

Utah State

Dynamic Text

- Can be modified at run-time.
- Update the `.text` property
- **Has** to have an instance name
- Example:
 - `faceState.text = "I'm sad!"`;

Utah State

ActionScript and Movie Clips

- Instance names
- "Pathing" to instances (not symbols):
 - `(root as variableType).instanceName.functionOrProperty`
 - (for absolute pathing)
 - For example: `(root as MovieClip).faceState.text = "sad"`;
 - `(parent as variableType).instanceName.functionOrProperty`
 - (for relative pathing)
 - For example: `(parent as MovieClip).faceState.text = "sad"`;
 - `this.instanceName.functionOrProperty`
 - (this refers to current location in heirarchy)
 - For example: `faceFrame = this.faceClip.nextFrame()`;
 - `this.` is usually not necessary, already implied

Utah State

Hierarchy example

```

betterB.addEventListener(...)
this.faceClip.nextFrame();
(parent as MovieClip).faceState.text = "Man, I'm sad";

```

Utah State

Another Hierarchy example

- From sky to mushroom:
 - (parent as MovieClip).mushroom
- From mushroom to sky: (parent as MovieClip).sky

Utah State
