Filthy Rich Flex Clients

Chet Haase

Flex SDK Team

November 17, 2008

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Agenda

MAX
CONNECT. DISCOVER. INSPIRE.

- Definition: Filthy Rich Clients
- Building Blocks of Filthy Rich Clients
- Filthy Flex Functionality
- Flexciting Flexamples

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Adobe

Definition

Filthy Rich Clients: *[Fýlthē Rýtczh Klìəñtż, Slang (Amer. Geek)]*

Applications that are so graphically rich that they ooze cool. They suck users in from the outset and hang onto them with a death grip of excitement. They make the users tell their friends about the application.

*They make the users actually **enjoy** the experience.*

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Who Needs to be Filthy?

- *Any* application can benefit
  - Doesn't have to use extreme (or annoying) animations
- Filthy rich techniques can improve usability of even simple applications
  - Subtle animations
  - Animated transitions
- Better user experience == more productive users
- Witness changes in standard computer desktops
  - Mac OS X, Vista
  - Many subtle animations and rich graphics everywhere

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Why Flex?

- Filthy rich capabilities come built-in with Flex
  - Rich graphics
  - Animation
  - Customizable

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Agenda

- Definition: Filthy Rich Clients
- Building Blocks of Filthy Rich Clients
  - Graphics
  - Image Processing
  - Animation
- Filthy Flex Functionality
- Flexciting Flexamples

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Graphics

- Graphics capabilities are at the heart of Filthy Rich Clients
  - Customized rendering for rich look & feel, unique experience
- Drawing Primitives + State
  - Shapes, lines, …
  - Colors, gradients, …
- Translucency
  - Blends source color (Cs) with destination color (Cd)

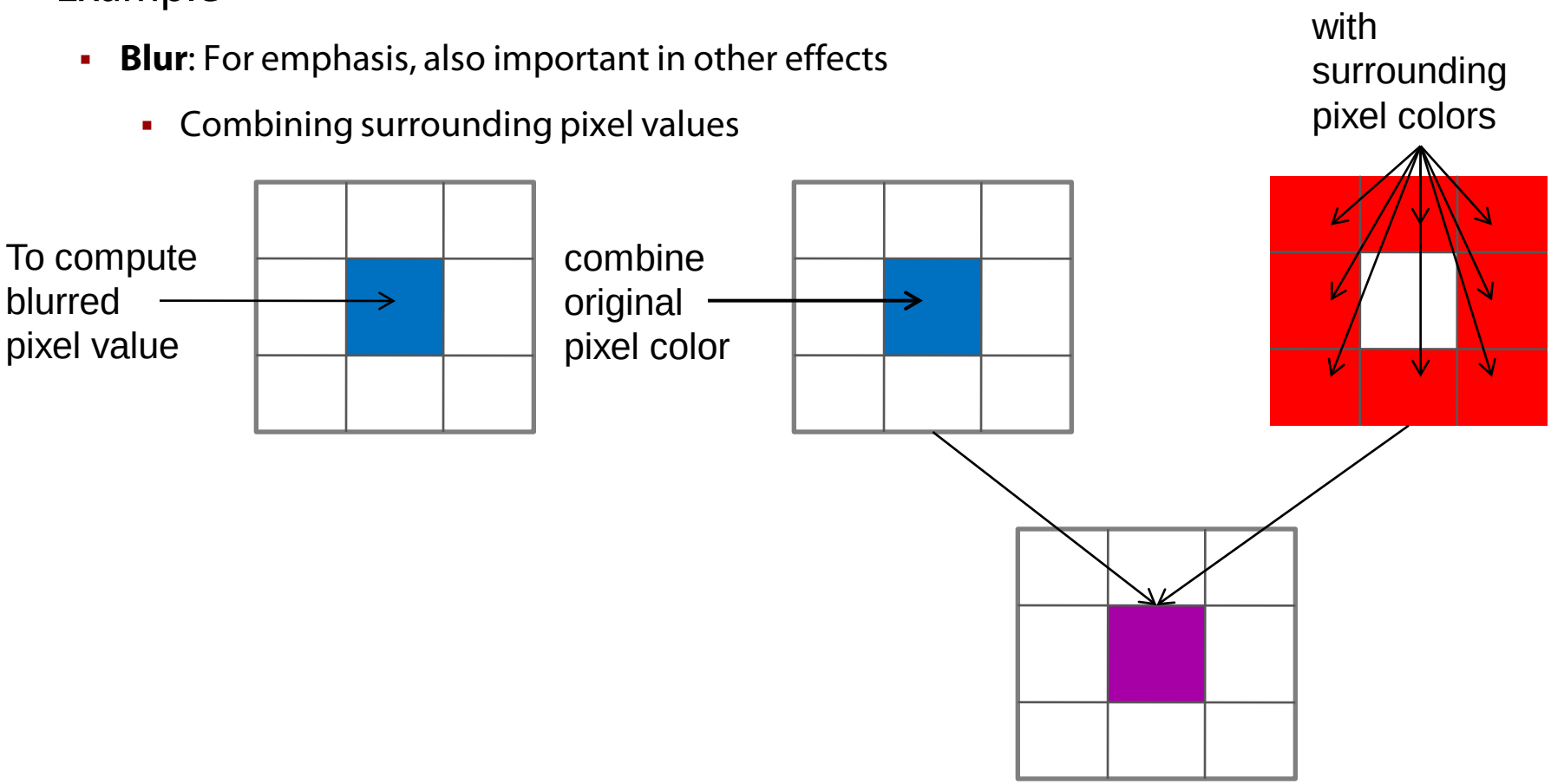
```
C = alpha*Cs + (1 – alpha)*Cd
```

- Images
  - Creation, display, manipulation
  - Media-rich applications
  - GUI Icons
  - Image-based effects

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Image Processing

- Altering the way that objects are displayed
- Example
  - **Blur**: For emphasis, also important in other effects
    - Combining surrounding pixel values



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Animation

- Animation: the change of value over time
  - Position, size, translucency, color, …
  - Examples:
    - Move button from current location to (100, 100) over .5 seconds
    - Pulsing glow on button that has focus
    - Progress bar
    - Fade components in/out during application state change
- Motion makes for dynamic applications
  - But don't go crazy…

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Agenda

MAX CONNECT. DISCOVER. INSPIRE.

- Definition: Filthy Rich Clients
- Building Blocks of Filthy Rich Clients
- Filthy Flex Functionality:
  - Graphics
  - Image Processing
  - Animation
- Flexciting Flexamples

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Adobe

Flex Graphics: Graphics Objects

- Flex components/objects *are* or *have* Flash display objects
  - Use Flash rendering operations to draw lines, paths, shapes
  - Don't forget graphics attributes: gradients, colors, strokes
- Tip: Don't forget to clear the display list!
  - Flash uses retained-mode rendering

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Graphics: Translucency

- All Flash (and Flex) display objects have **alpha** property
- To alter the translucency of a Flex component, just set alpha property
- Demo: Translucency

```
<mx:Image width="60%" horizontalCenter="0"
    source="@Embed(source='assets/b.jpg')"
    alpha="{slider.value}"/>
<mx:Button x="378" y="186" label="Disappearing Button"
    verticalCenter="0" horizontalCenter="0“
    alpha="{slider.value}"/>
<mx:HSlider id="slider" x="10" y="412"
    width="100%" bottom="10” liveDragging="true“
    minimum="0" maximum="1" value="1"/>
```

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Graphics: Image Support

- Flash and Flex have plenty of image support built-in
- Loading

```
<Image source="path"/>
```

- Creating from scratch

```
var bmData:BitmapData =
    new BitmapData(w, h, transparent, fillColor);
var bitmap:Bitmap = new Bitmap(bmData, pixelSnap, smoothing);
```

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Image Processing: Filters

- Filters are simple image processing operations
  - Parameterized for flexibility
- Examples:
  - BlurFilter: Blurring contents of a display object
  - Glow: Blur+tint over component
  - Drop shadow: offset glow
- Demo: Filters

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation: Timer

- Most basic of animation APIs
- Callback system
  - You tell it how often to call your handler
  - It calls the handler at that frequency (or as close as possible)
- Application code handles all of the 'animation' details
  - Calculating elapsed time
  - Calculating values based on time
  - Setting values on target objects

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation: Tween

- Core animation class
  - Takes animation parameters, start/end values, listeners
  - Runs **`Timer`** internally
  - Calls listeners with new/calculated values during animation
  - Optional easing capability for more realistic/interesting motion
- Caller associates values with target objects/properties
  - Tween is target/property agnostic

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation: Effects

- Higher-level animation objects
- Associate animation parameters with target object and properties
  - **Move** animates x/y properties on target
  - **Fade** animates alpha on target
- Also add extra intelligence
  - **Fade** animates alpha when `visible` property changes between states
  - Integration with Transitions

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation: Transitions

- Associate effect(s) with state changes
- Supply properties/values to effects, based on state information
  - **Move** gets before/after x/y values
  - **Fade** gets visibility/alpha/existence values
- Transition effects automatically run when **`currentState`** changes

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

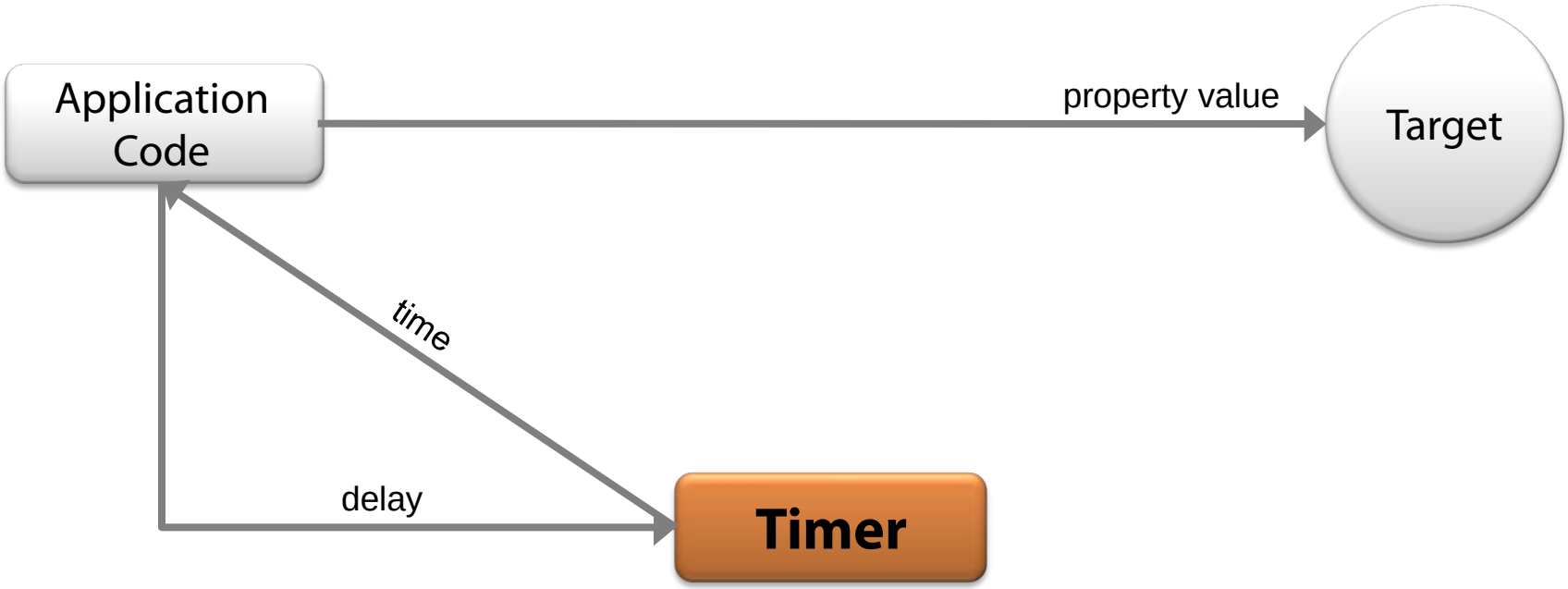
Flex Animation: Triggers

- Like transitions, but based on events
- Automatic setup/run of effects when certain things happen
  - **`moveEffect="Move"`** runs Move effect automatically when target's location changes
    - x/y information supplied to Move by event information

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation, the Diagram: Timer

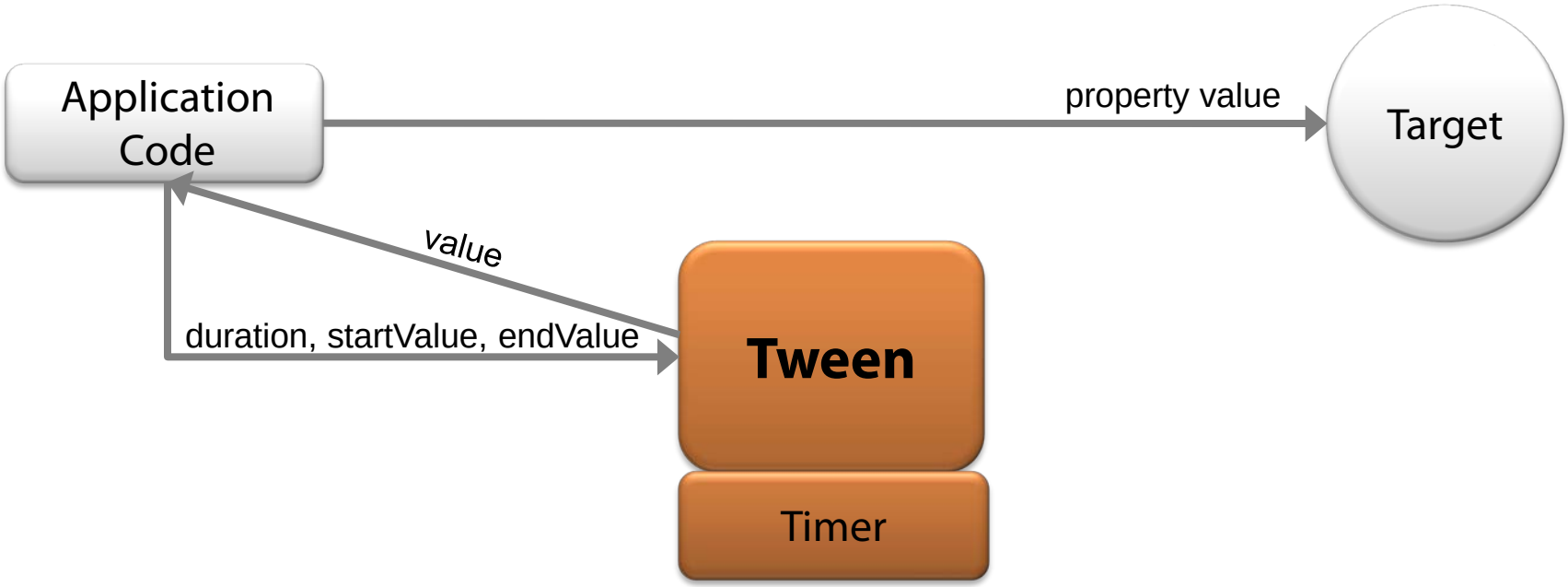
```
var timer:Timer = new Timer(delay);
timer.addEventListener(TimerEvent.TIMER, timerHandler);
timer.start();

public function timerHandler(event:TimerEvent) {
    // calculate value based on new time, set target properties
}
```



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation, the Diagram: Tween

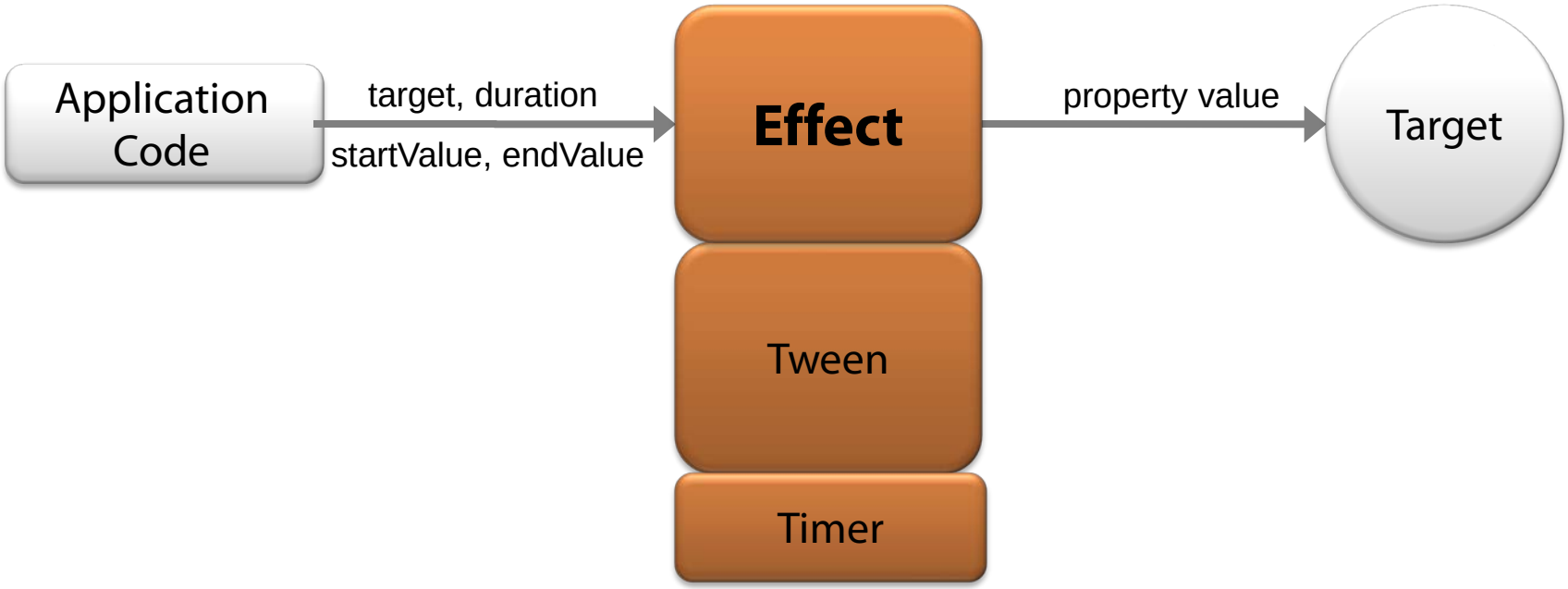
```
var tween:Tween = new Tween(this, startVal, endVal, duration);

public function onTweenUpdate(value:Object) {
    // set target property to value
}
```



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation, the Diagram: Effect

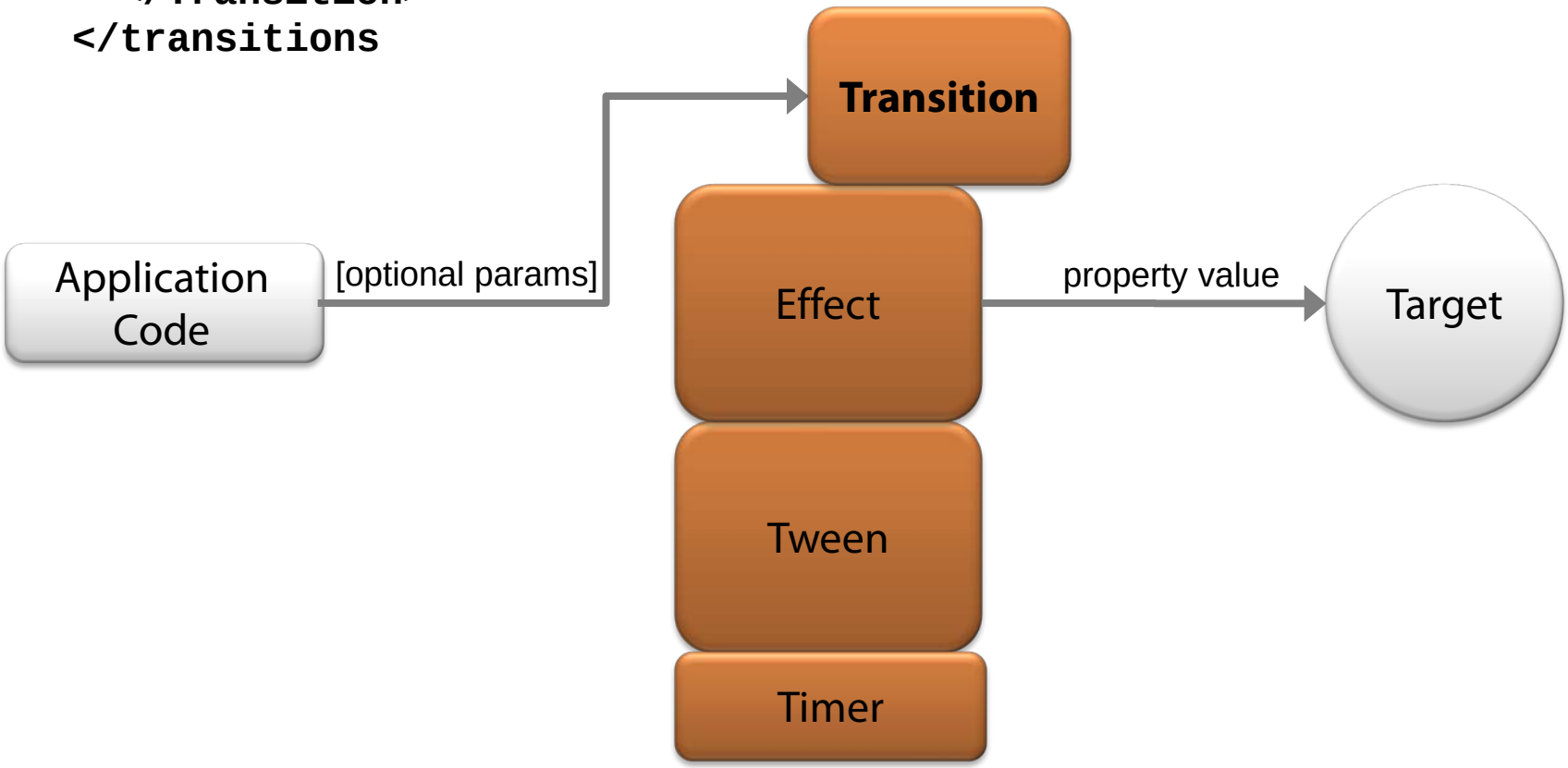
```
<Fade id="fader" target="{someTarget}" duration="200"
    alphaFrom="1" alphaTo="0"/>
<Button click="{fader.play()}"/>
```



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation System: A Graphical View

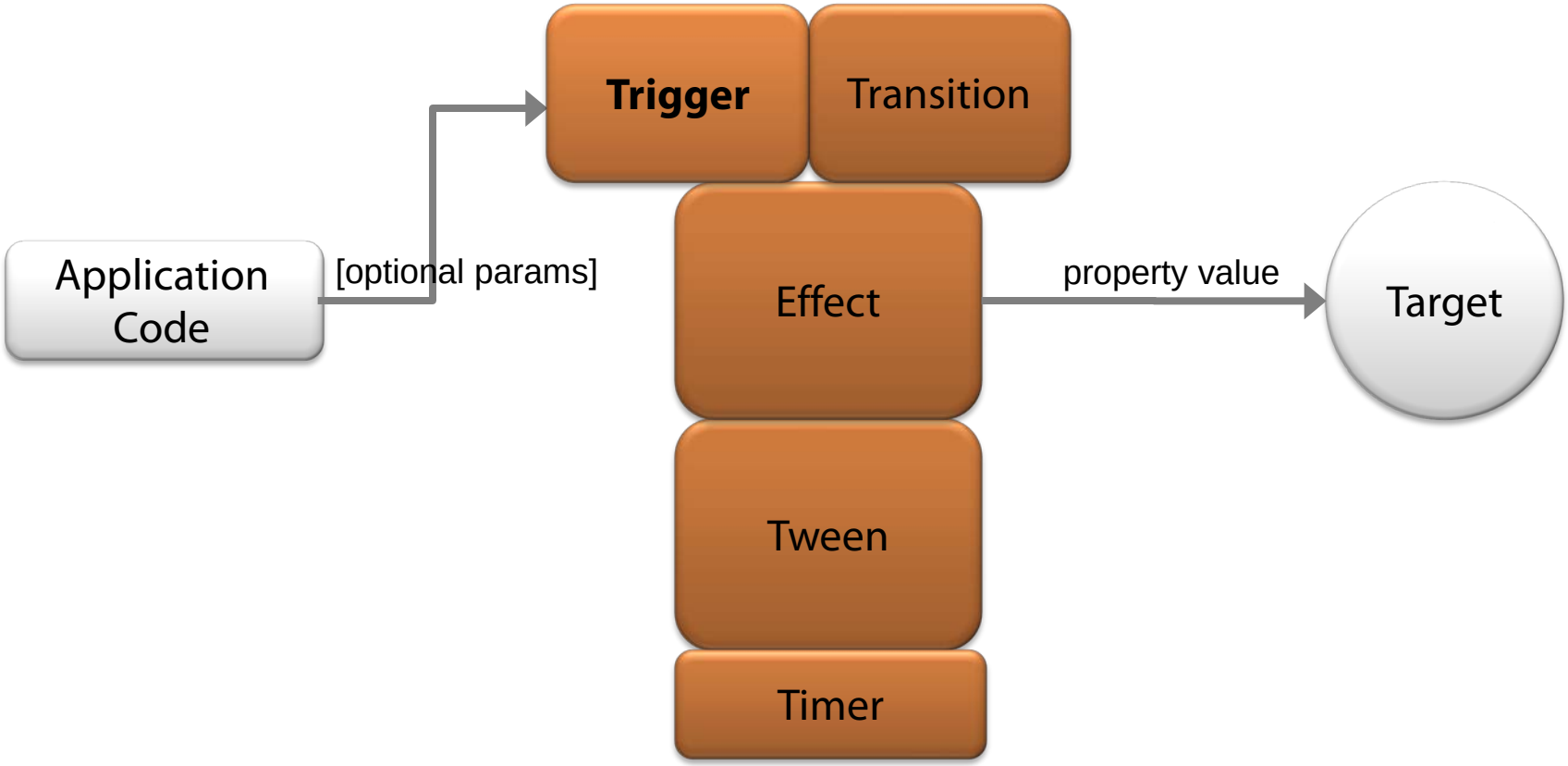
```
<!– assume states definition -->
<transitions>
  <Transition>
    <Fade targets="{[myTarget]}"/>
  </Transition>
</transitions
```



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Flex Animation System: A Graphical View

```
<Fade id="fader" duration="200" alphaFrom="1" alphaTo="0"/>
<Button removedEffect="{fader}"/>
```



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

The Filthy Future!

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Agenda

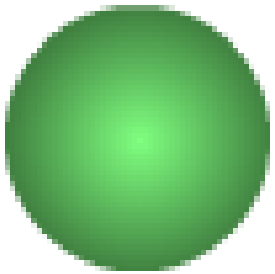
- Definition: Filthy Rich Clients
- Building Blocks of Filthy Rich Clients
- Filthy Flex Functionality: Gumbo
  - Graphics primitives
  - Flex Filters
  - Animation
- Flexciting Flexamples

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Graphics

- FXG: Flex graphics primitives
  - Declarative objects
  - For component skinning or arbitrary graphics rendering
  - No more Flash drawing commands to add shapes to the DOM
  - No need for custom UIComponent to add DisplayObject children to component

```
<Ellipse width="50" height="50" id="ball">
  <fill>
    <RadialGradient>
      <entries>
        <GradientEntry color="0x80ff80" ratio="0"/>
        <GradientEntry color="0x408040" ratio="1"/>
      </entries>
    </RadialGradient>
  </fill>
</Ellipse>
```



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Filters

- Flex versions of Flash filters
  - E.g., GlowFilter, BlurFilter, …
  - Automatically update the component owner when filter properties change
- Pixel Bender (Flash Player 10)
  - Cross-platform shading language for image filtering operations
  - Flex 'ShaderFilter': apply shader as filter on components
  - FxAnimateFilter: animate properties of ShaderFilter (or other Flex filters)
- Demo: ShaderLand

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation

- Animating arbitrary objects
- Animating arbitrary types
- API improvements
  - Animation object
  - API changes
  - New functionality
- More built-in Effects
- More intelligent Effects

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation: Animating Arbitrary Objects

- Flex 3 effects animate targets in various ways
  - Move calls target.move(x, y)
  - Resize sets target.width/target.height
  - Rotate sets target.rotation and called target.move(x, y)
  - Difficult to generalize effects for different target types
- Gumbo effects animate named properties on targets
  - FxMove sets target["x"], target["y"]
  - FxResize sets target["width"], target["height"]
  - FxRotate sets target["rotation"], target["x"], target["y"]
  - FxAnimate superclass can animate any named property/style
    - To move 'button' from (0,0) to (100, 200):

```
<FxAnimate id="mover" target="{button}">
  <PropertyValuesHolder property="x" values="{[0, 100]}"/>
  <PropertyValuesHolder property="y" values="{[0, 200]}"/>
</FxAnimate>
```

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation: Animating Arbitrary Types

- Flex 3 effects only understand **`Number`**
  - Works for most situations (x, y, width, height, rotation, alpha, etc.)
  - But not for all (colors, arbitrary objects)
- Gumbo effects work with arbitrary types instead
  - Default is still Number
  - Effects can supply their own type interpolator
    - FxAnimateColor supplies a ColorInterpolator
  - Developers can supply their own interpolator to the effects as well

```
public class FooInterpolator {
  public function interpolate(start:*, end:*, fraction:Number):void
  {
    var val:Foo;
    // Code to interpolate Foo type between start and end, given
    // the elapsed fraction
    return val;
  }
}
```

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation: API Improvements: Animation

- **`Animation`** is the new **`Tween`**
- Low-level timing engine that Gumbo effects use
- Can also use Animation directly
- Takes animation attributes, start/end values, updates listeners with animated values

```
// Moves object foo from x=0 to x=100 over a half-second
var anim:Animation = new Animation(0, 100, 500);
anim.addEventListener(AnimationEvent.ANIMATION_UPDATE, updater);
anim.play();

private function updater(event:AnimationEvent):void {
  foo.x = event.value;
}
```

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation: API Changes (since Tween)

- No more auto-play on construction
  - Can pre-construct and call play() when needed
- Only event listeners
  - No more callback handlers and implicit Tween updateHandler/endHandler functions
- Events for start/end/update/repeat
- New IInterpolator and IEaser interfaces
  - Arbitrary types, parameterized easing, easy API for custom implementations

```
anim.easer = new Sine(.7); // Sine, ease in for first 70%
anim.easer = new Power(.4, 3); // Cubic, ease-in for first 40%
anim.easer = new Power(1, 4); // Quartic, ease-in over entirety

public class ReverseEase implements IEaser {
  public function ease(fraction:Number):Number {
    return (1 – fraction);
  }
}
```

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation: API Functionality

- Repetition, including new 'Reverse' behavior
- Single timing pulse for all timing events
  - Easier to synchronize multiple effects with delays

```
// 'bouncing' behavior
var anim:Animation = new Animation(0, 300, 1000);
anim.easer = new Power(1, 3);            // ease in all the way
anim.repeatBehavior = Animation.REVERSE; // reverse every time
anim.repeatCount = 0;                    // forever
anim.play();

// animation sequencing
var anim:Animation = new Animation(100, 200, 500)
var anim1:Animation = new Animation(200, 300, 1000);
// anim1 will start in same timing pulse as anim ends
anim1.startDelay = 500;
anim.play();
anim1.play();
```

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation: New Effects

- FxAnimateColor
  - auto-tween a color between two colors over time
- FxAnimateFilter
  - animate a Pixel Bender shader or other filter over time
- FxAnimateRotate3D
  - animate 3D transform properties
- CallAction
  - call arbitrary function, useful for sequencing with other effects
- SetAction
  - replaces SetPropertyAction/SetStyleAction
- Pause
  - now waits for timeout *or* arbitrary events
- (and we're not done yet…)

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Animation: More Intelligent Effects

- Fade-without-Remove
- In Flex 3, had to use RemoveAction to fade objects in and out of existence:

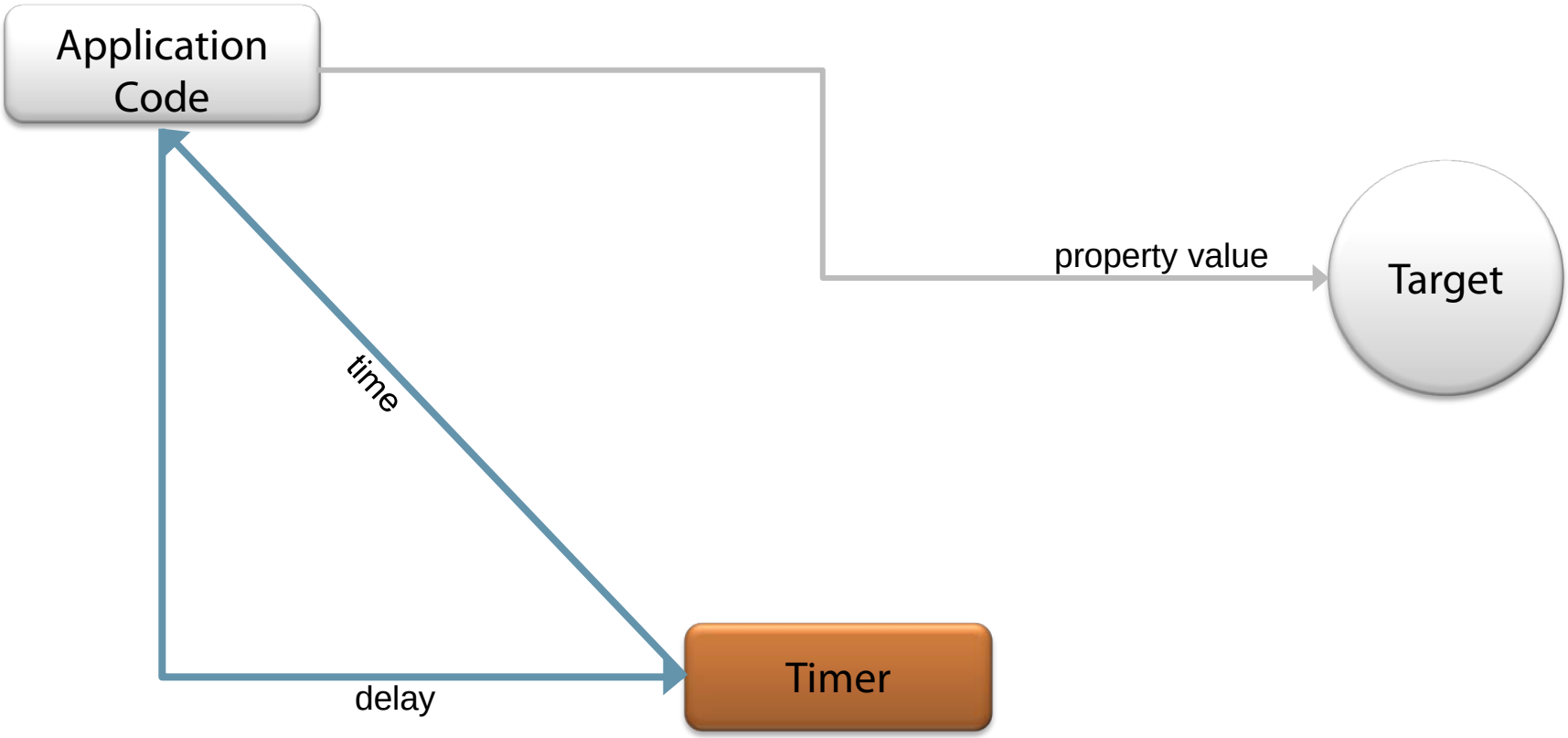
```
<Transition fromState="A" toState="B">
  <Sequence target="{button}"/>
    <Fade/>
    <RemoveAction/>
  </Sequence>
</Transition>
```

- Gumbo takes care of this for you:

```
<Transition fromState="A" toState="B">
  <Fade target="{button}"/>
</Transition>
```

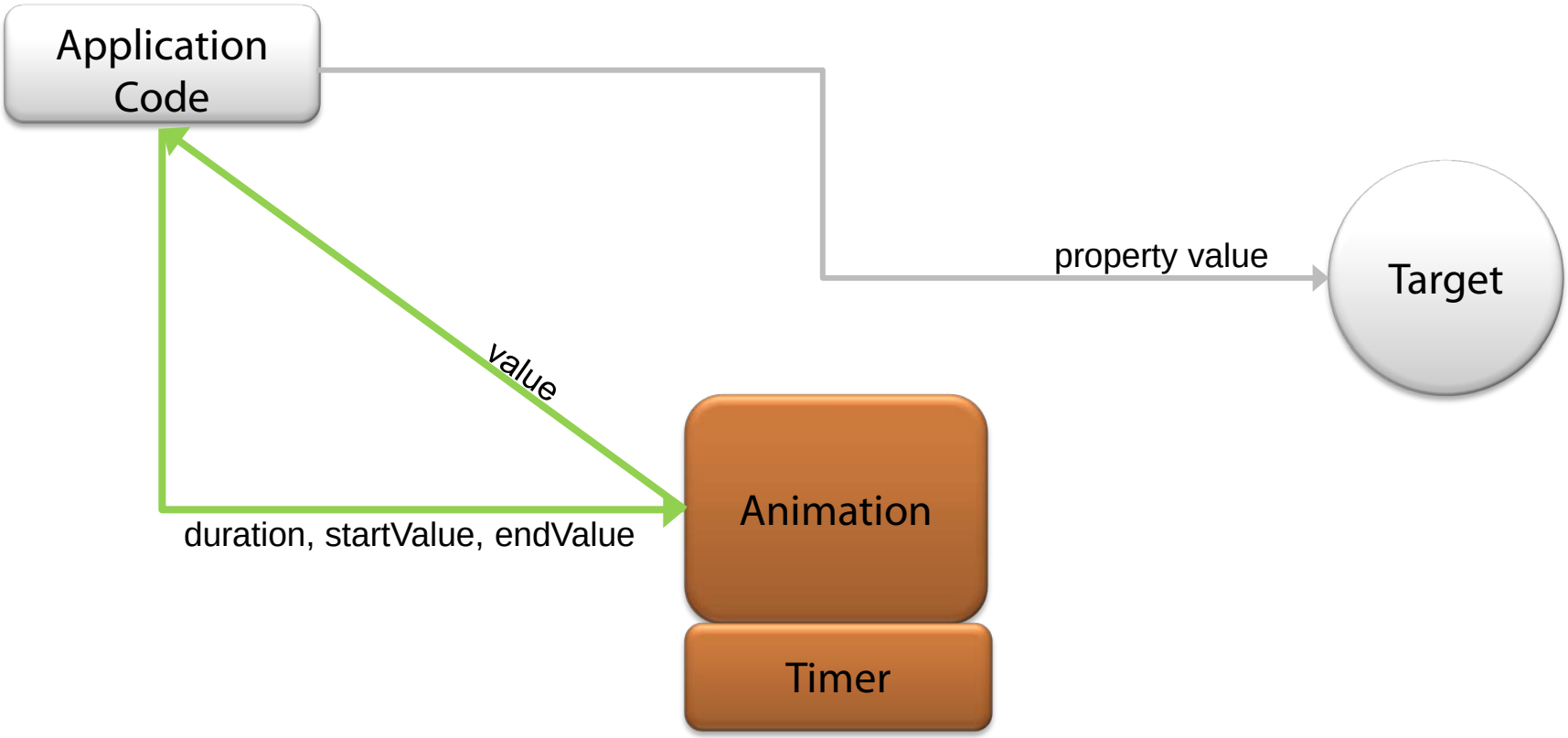
Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Effects: The Helpful Diagram



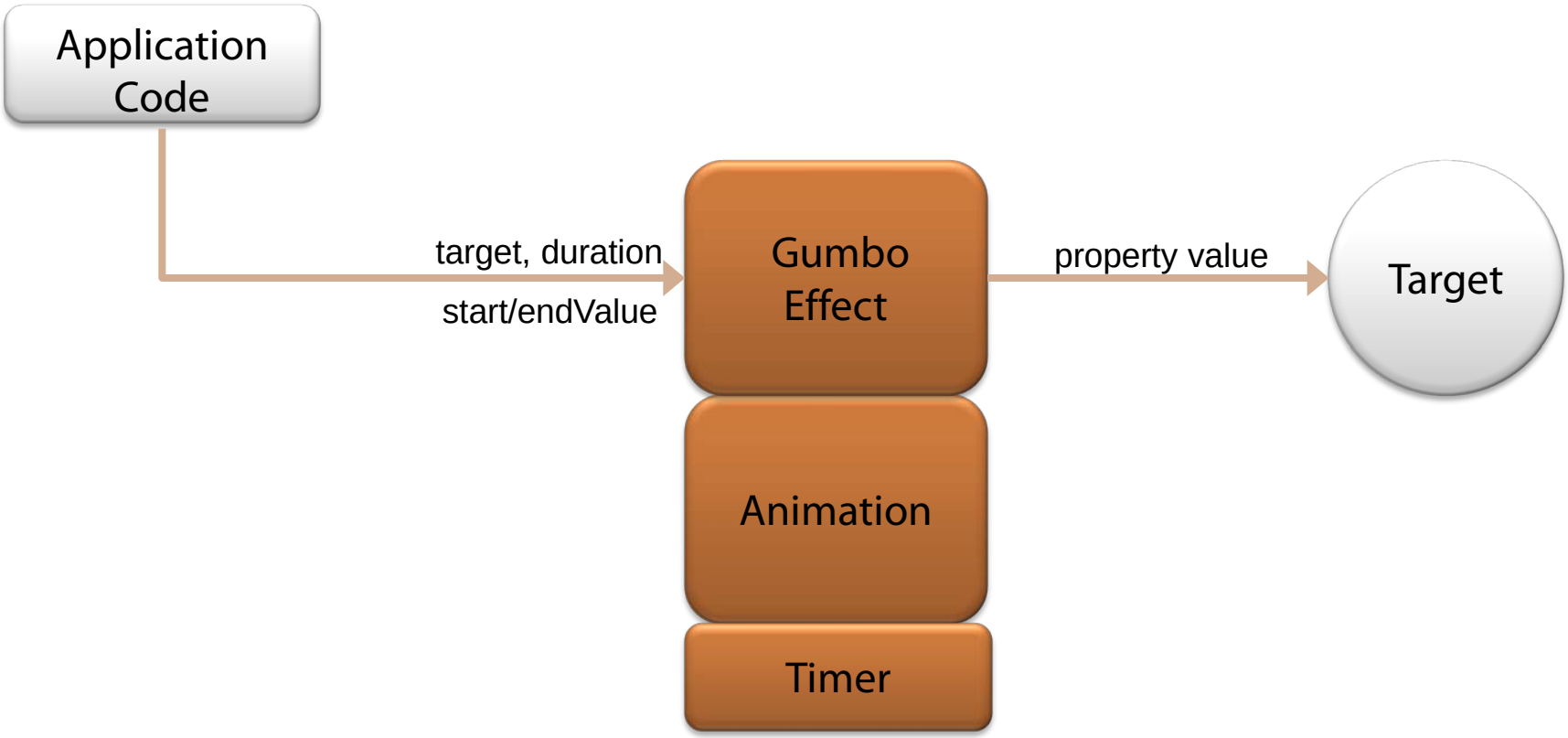
Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Effects: The Helpful Diagram



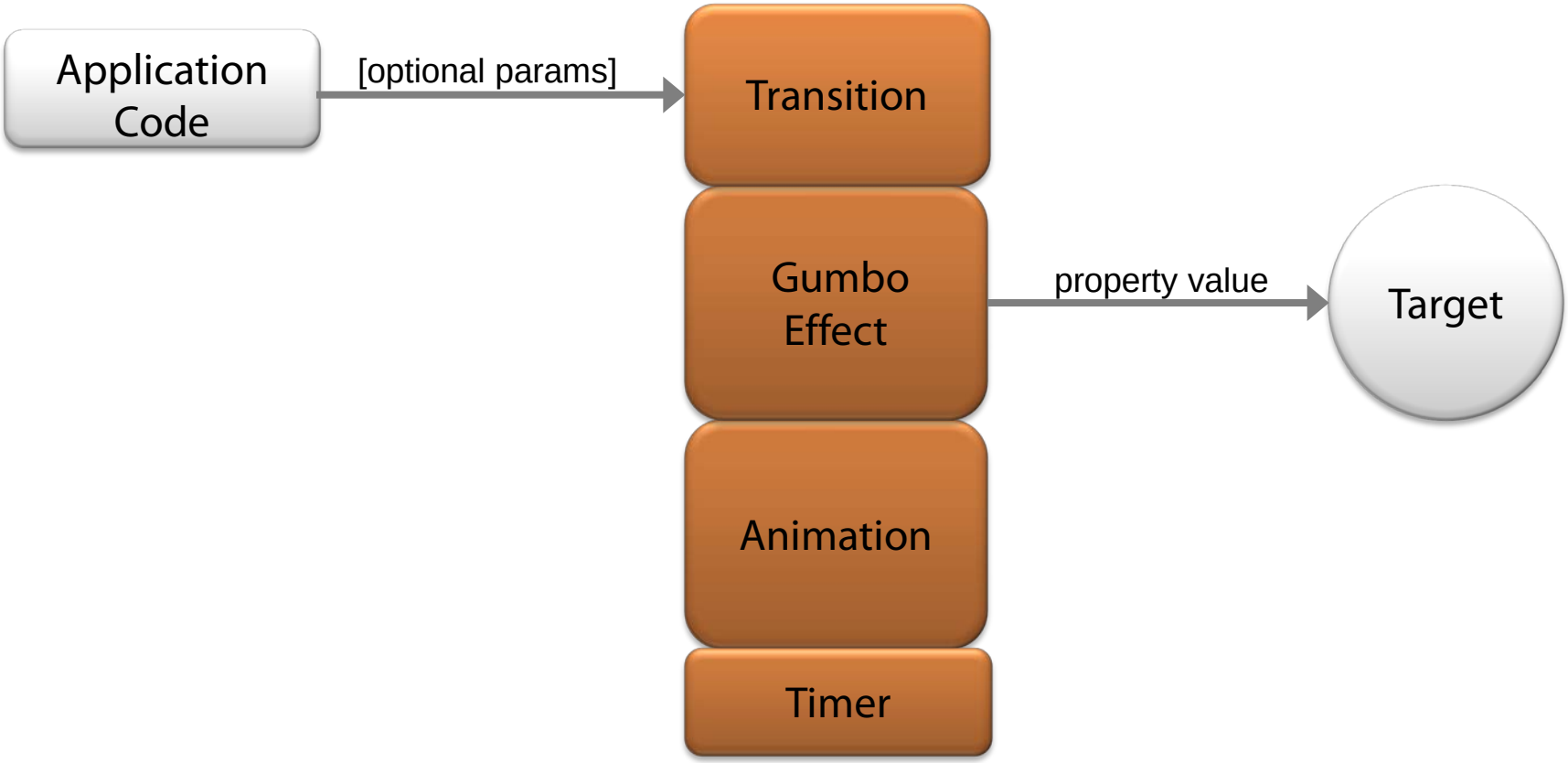
Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Effects: The Helpful Diagram



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Gumbo Effects: The Helpful Diagram



Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Animation Demos

- Coolerizer
  - Arbitrary type interpolation
  - Ne w effects
  - New Animation capabilities
  - Nifty colors
- EasingTest
  - Various animation controls
  - New easing functionality
- DynamicMenus
  - Reversing transitions

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Agenda

MAX
CONNECT. DISCOVER. INSPIRE.

- Definition: Filthy Rich Clients
- Building Blocks of Filthy Rich Clients
- Filthy Flex Functionality
- Flexciting Flexamples

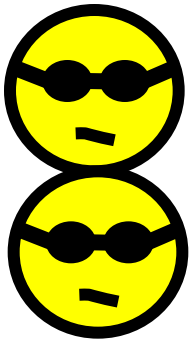
Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Adobe

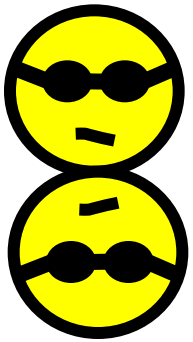
Reflection

- What
  - Reflection container
  - Add component/DisplayObject, it displays that plus reflection
- Why
  - 3D-like effect
  - Very rich presentation style
- How
  - Add input DisplayObject

  - Make a bitmap copy



  - Inverse-scale it

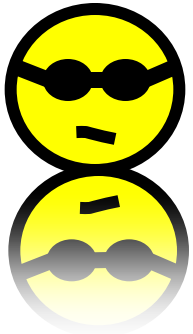


Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Reflection: How

- Create an alpha-gradient

- Use the gradient as a mask



- Limit the length of the mask

- Blur the result

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Dynamic Drag & Drop

- What
  - Show item being dragged, animate results
- Why
  - Visual feedback on operations
  - Transitions to new states
- How
  - Dragged object visual: Create DragSource with Image proxy
  - Differentiation: Set translucent alpha on proxy
  - Animate image change with 2 Fade effects
  - Flex handles the rest
    - Cues on allowable drop targets
    - Animations back to the source or onto the destination

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Animated Transitions: SearchTransition

MAX CONNECT. DISCOVER. INSPIRE.

- What
  - Search query and results pages
  - Smooth transition between states
- Why
  - Avoid disruptive UI changes
  - User can lose their place when UIs are abruptly changed
- How
  - Set up different states (FlexBuilder and/or MXML code)
  - Declare transition effects
  - Use triggers for some effects

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Adobe

Animated Transitions: SlideSorter

- What
  - Picture album viewer
  - Resizing thumbnails animates slides to new places/sizes
- Why
  - Avoid disruptive UI changes
  - User can lose their place when UIs are abruptly changed
- How
  - Standard Resize/Move effects for animations on pictures
  - Custom transition
    - Capture start values
    - Validate layout, then disable
    - Play effect
    - Re-enable layout

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Pulse

- What
  - Icon with Animating glow effect
- Why
  - Subtle, dynamic indication to user
  - Get their attention without annoying them
- How
  - Add Flash GlowFilter to image component
  - Animate filter with AnimateProperty
  - Re-establish changed filter on component for each update
  - *Gumbo note: New Flex filters will handle changes automatically*

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Spring

- What
  - Fade/Zoom effect when icon is double-clicked
- Why
  - Give the user indication that the system is processing their request
- How
  - Custom zoom effect
    - copy component into Bitmap
    - Scale bitmap around center
    - Fade bitmap alpha to 0

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Related MAX Sessions

MAX CONNECT. DISCOVER. INSPIRE.

- If you liked [this session] you might also like
  - Tonight 8:30: Meet the Flex Team
  - Tuesday 1:30 Creating Effects with Pixel Bender (lab)
  - Tuesday 1:30 Creating New Components in Flex 3 and Beyond

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Adobe

For More Information

- My Flex blog: http://graphics-geek.blogspot.com
  - Demos, code, video tutorials, animated discussion
  - Some of these demos posted already, the rest will show up eventually
- Gumbo
  - http://opensource.adobe.com/wiki/display/flexsdk/Flex+SDK
  - Buildable source of ongoing project
- Flexy
  - Flex-less version of Animation engine
  - No effects, but has Animation class, interpolation, easing capabilities
  - Not supported product: posted for quick Gumbo-agnostic intro to animation changes
  - http://graphics-geek.blogspot.com (search for Flexy)
- MAX sessions
  - Meet the Flex team (tonight 8:30), Creating New Components (Tuesday 1:30), …

Copyright 2008 Adobe Systems Incorporated. All rights reserved.

Filthy Rich Flex Clients

Chet Haase

Flex SDK Team

November 17, 2008

Copyright 2008 Adobe Systems Incorporated. All rights reserved.