

Macoun

Animationen für den Alltag

Clemens Wagner

Core Animation

=

Layer

+

Animationen

Layer

Stellen Inhalte dar

Modell
(Daten)

Präsentation
(aktueller Zustand)

Rendering
(privat)

KVC-Container

```
// Deklaration
@property (nonatomic) CGFloat depth;

// Implementierung
@dynamic depth;

// Zugriff
self.depth = 5.0; // Property / Autoboxing
NSNumber *theDepth = [self valueForKey:@"depth"]; // KVC
```

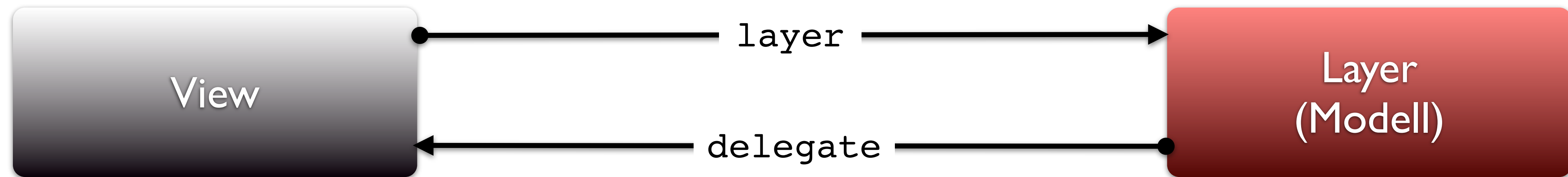
- Initialwert: +defaultValueForKey:

Layer neuzeichnen

- Neuzeichnen bei Wertänderung

```
+ (id)needsDisplayForKey:(NSString *)inKey {  
    return [inKey isEqualToString:@"depth"] ||  
           [super needsDisplayForKey:inKey];  
}
```

Layer und UIViews



```
// im View: Festlegen der Layer-Klasse
+ (Class)layerClass {
    return [KochLayer class];
}
```

Layer-Inhalt zeichnen

- Zeichnen über Core Graphics
- Subclassing `drawInContext:`
- Delegation

Inhalt über View zeichnen

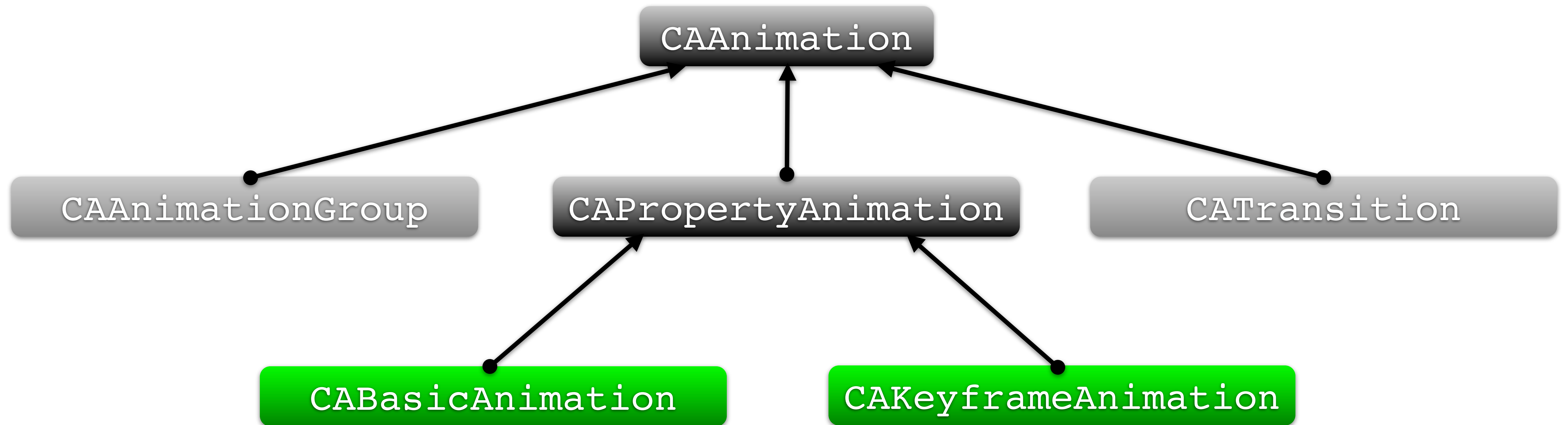
```
- (void)drawLayer:(CALayer *)inLayer inContext:(CGContextRef)inContext {  
    // hier malen  
}  
  
- (void)drawRect:(CGRect)inRect {  
    // bleibt leer  
}
```

Demo:

Layer mit View

Animationen

Animationsklassen



CABasicAnimation

Interpolation: `fromValue`, `byValue` und `toValue`

CABasicAnimation

Interpolation: `fromValue`, `byValue` und `toValue`

`fromValue` ●————→ `toValue`

`fromValue` ●————→ `fromValue + byValue`

`toValue - byValue` ●————→ `toValue`

`fromValue` und `toValue` ggf. auch aus Präsentations-Layer

CAKeyframeAnimation

- „viele“ Basic-Animations
- Interpolation mit mehreren Werten (`NSArray values`)
- Animation entlang einer Kurve (`CGPathRef path`)

bounds opacity position
anchorPoint center
backgroundColor contents
transform

transform.translation.y
transform.scale.x
sublayerTransform.rotation.z

Was kann ich animieren?

CGColor CGImage NSNumber

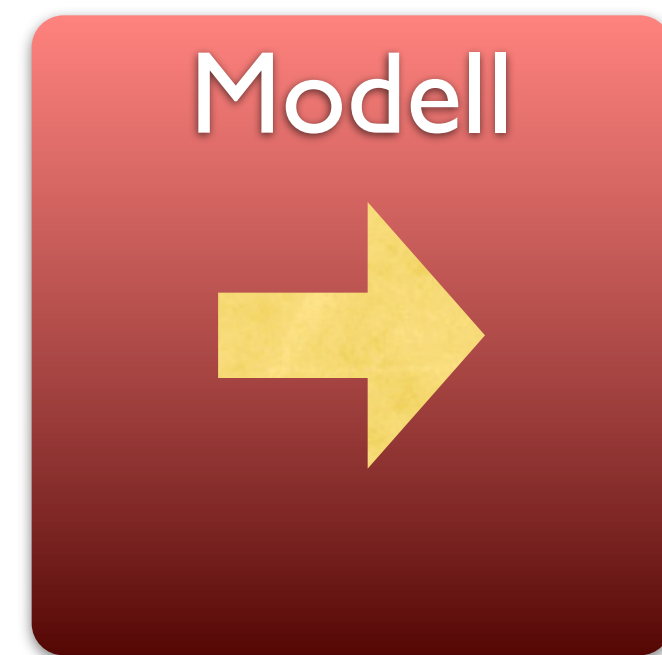
NSNumber mit

CGPoint, CGSize, CGRect, CGAffineTransform, CATransform3D

Animationsablauf

- Animation erzeugen
- Zum Layer hinzufügen: `addAnimation(forKey:`
- Ablauf
- Animationsende *automatisch* oder `removeAnimationForKey:`

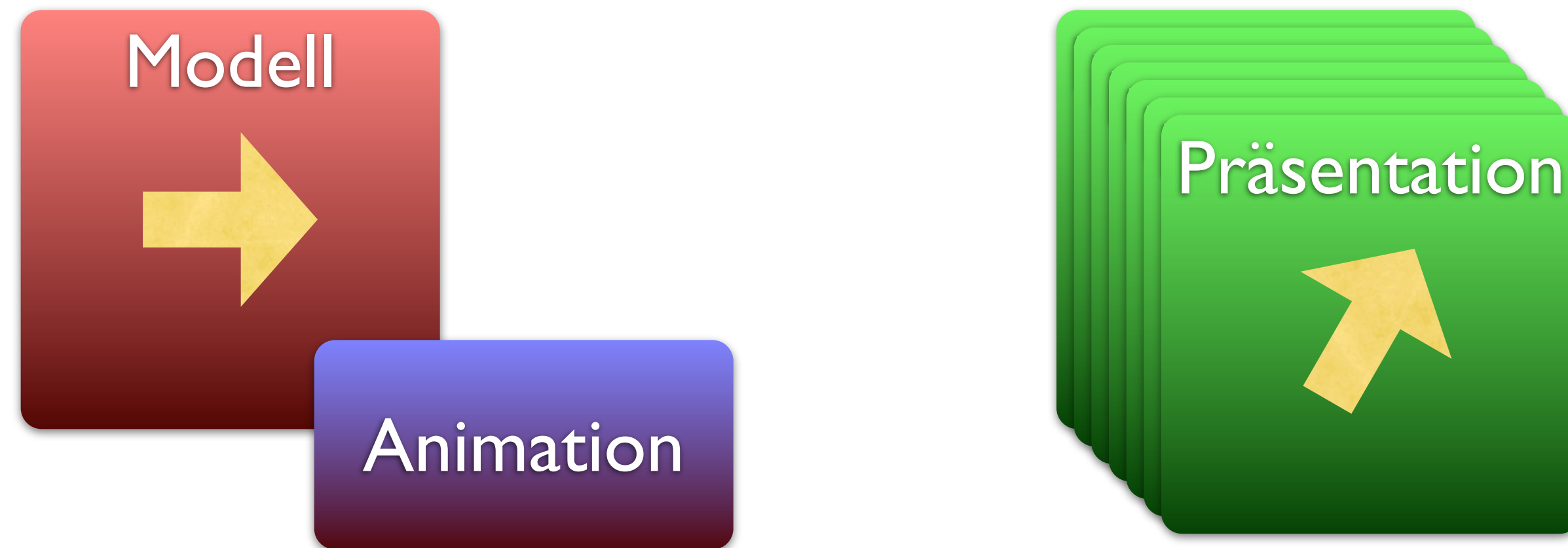
Animationsablauf (naiv)



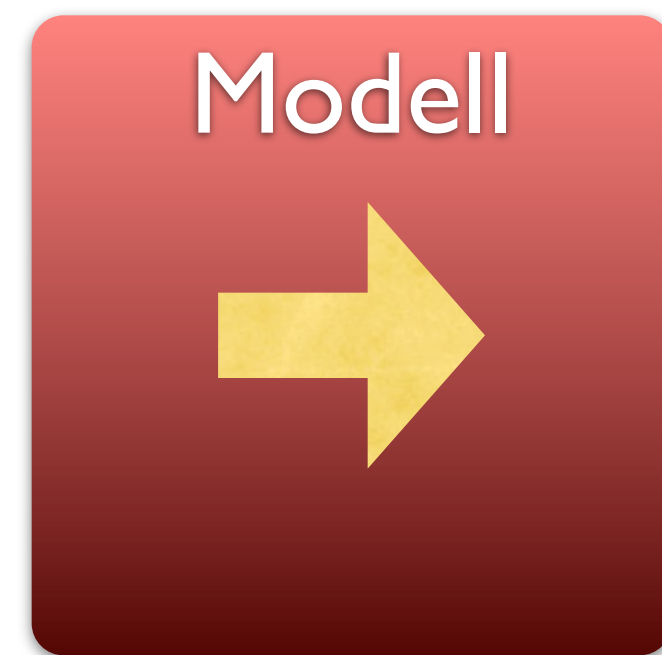
Animationsablauf (naiv)



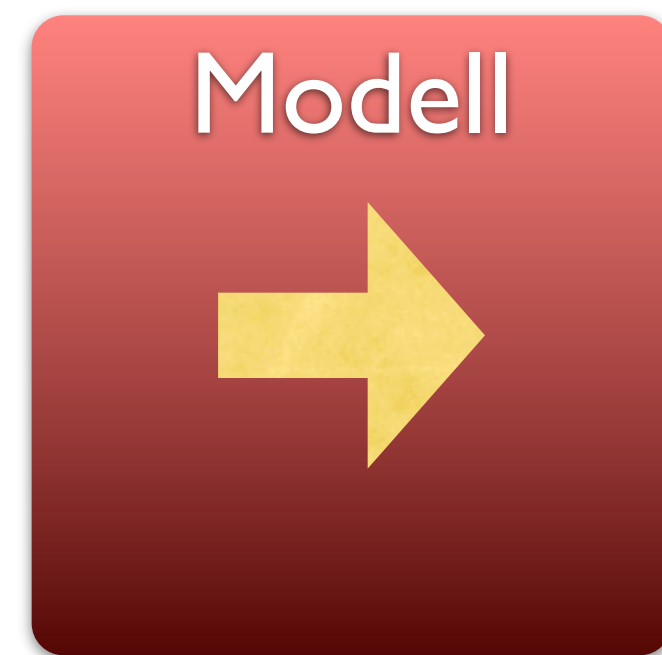
Animationsablauf (naiv)



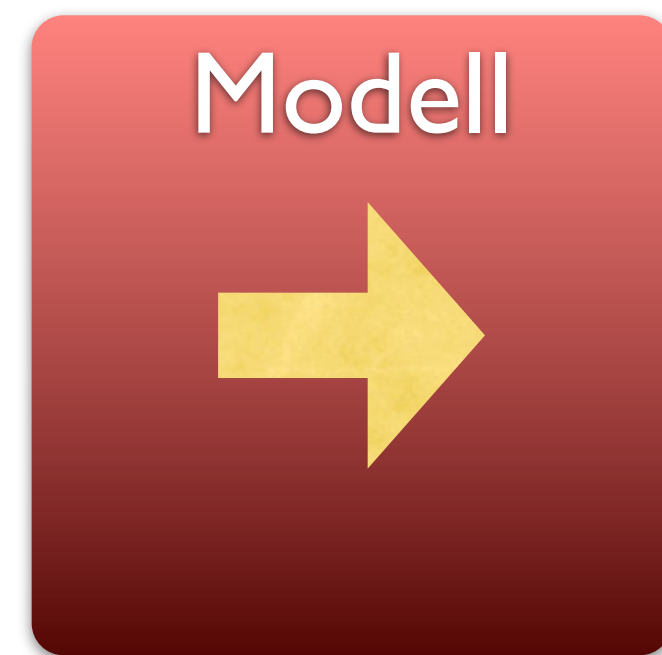
Animationsablauf (naiv)



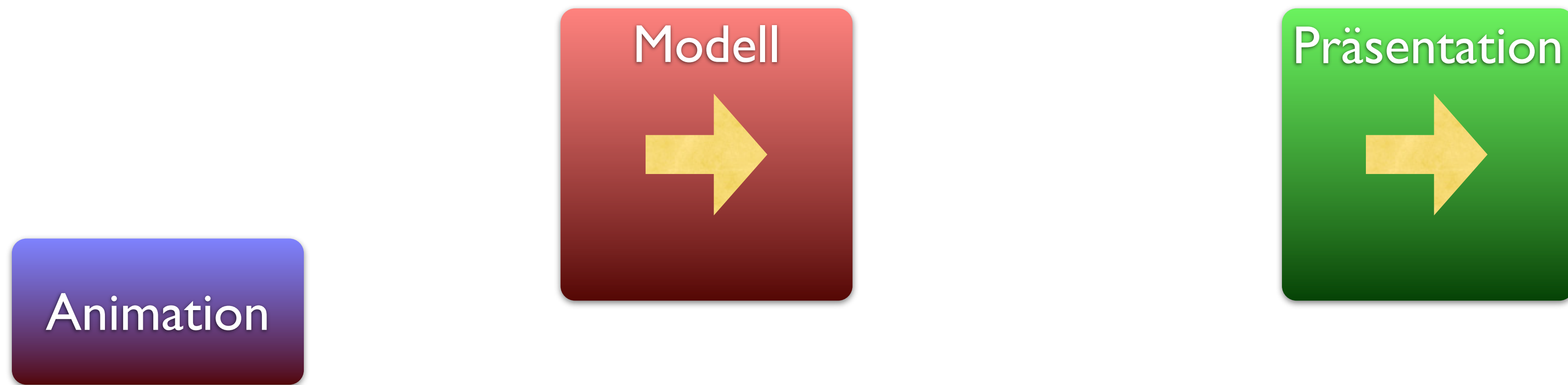
Animationsablauf (naiv)



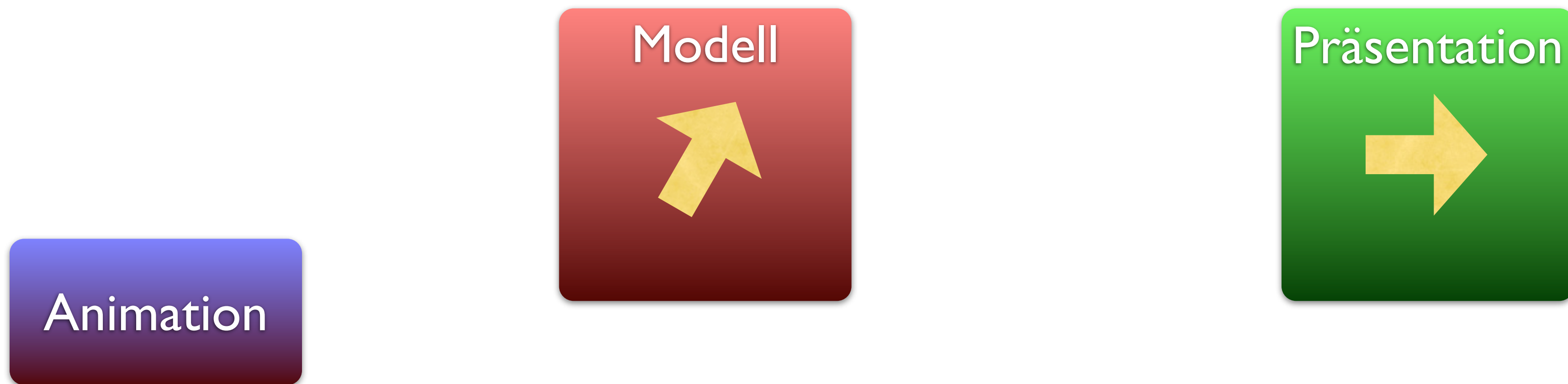
Animationsablauf (besser)



Animationsablauf (besser)



Animationsablauf (besser)



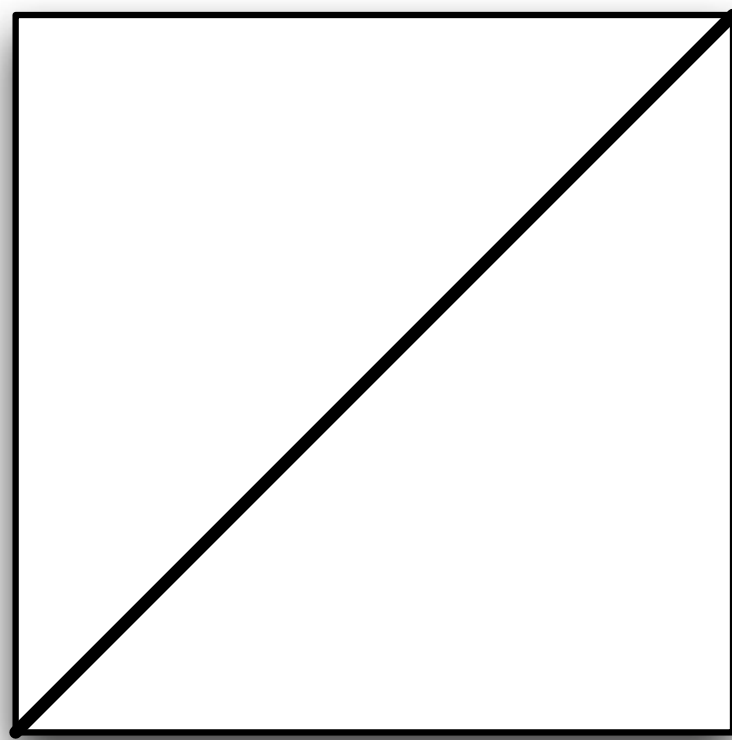
Animationsablauf (besser)



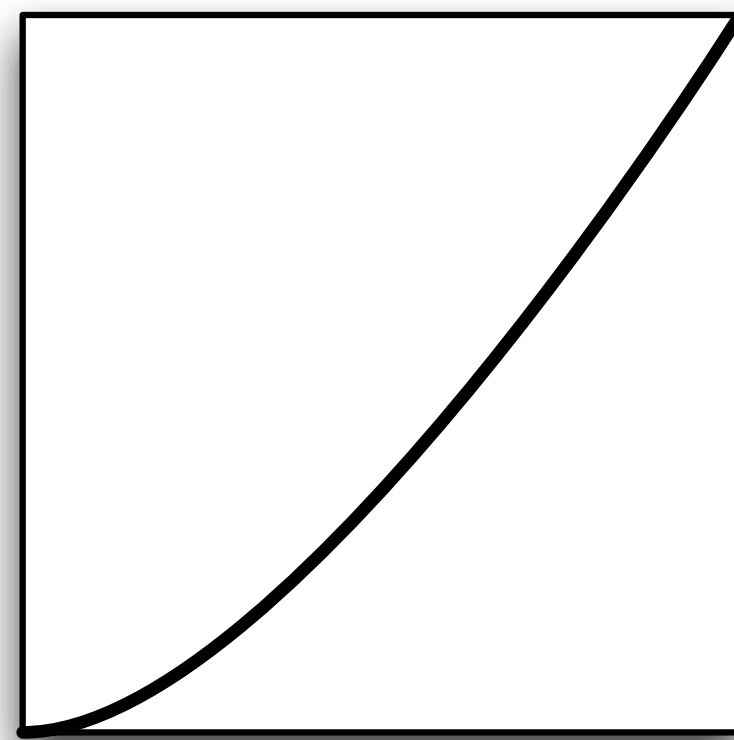
Animationsablauf (besser)



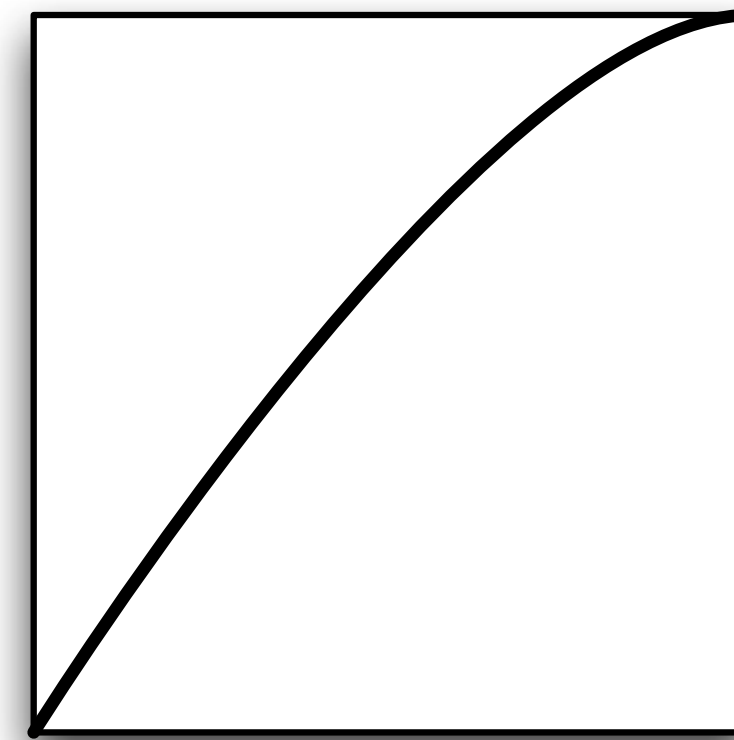
Standard-Timing-Funktionen



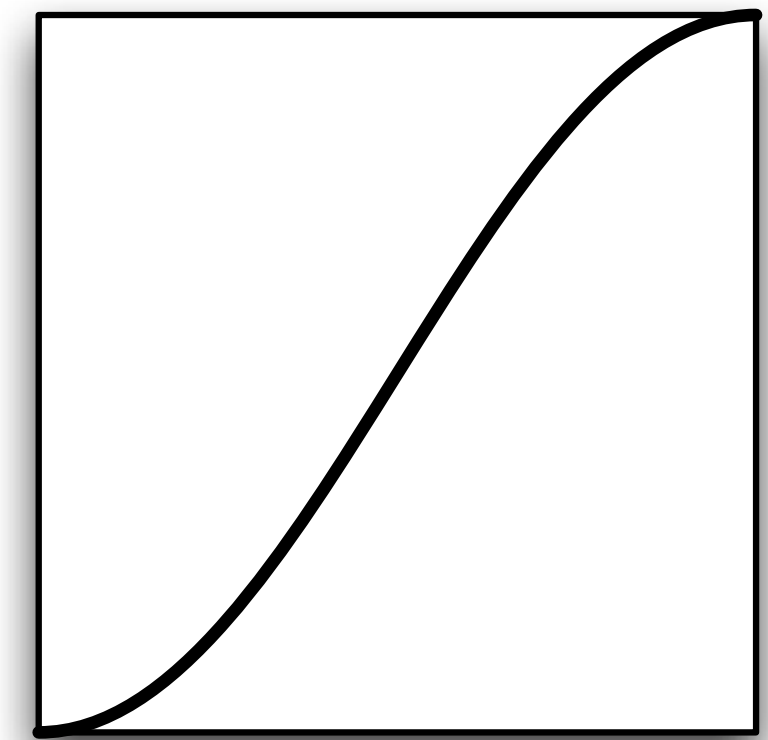
Linear



Ease-In

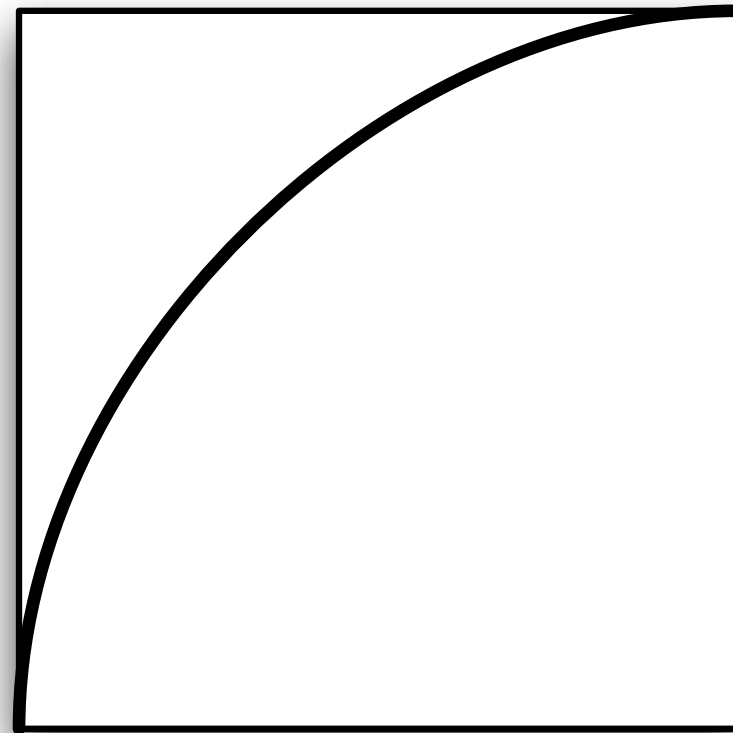


Ease-Out

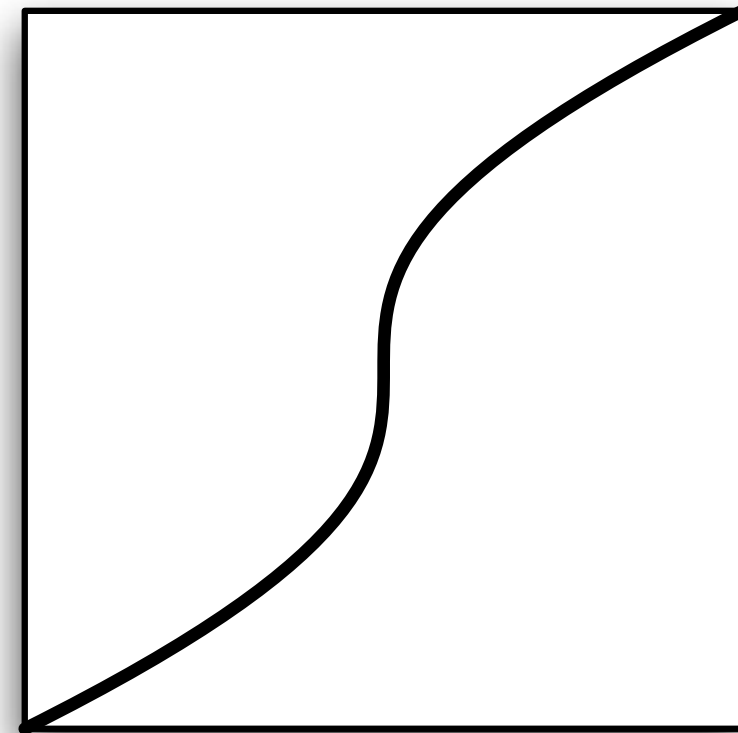


Ease-In-
Ease-Out

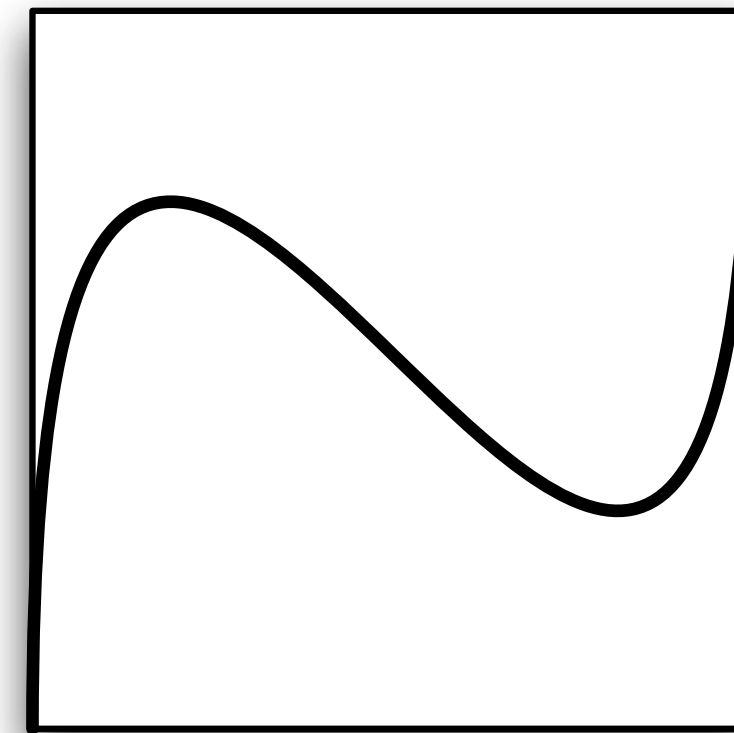
Benutzerdefinierte Funktionen



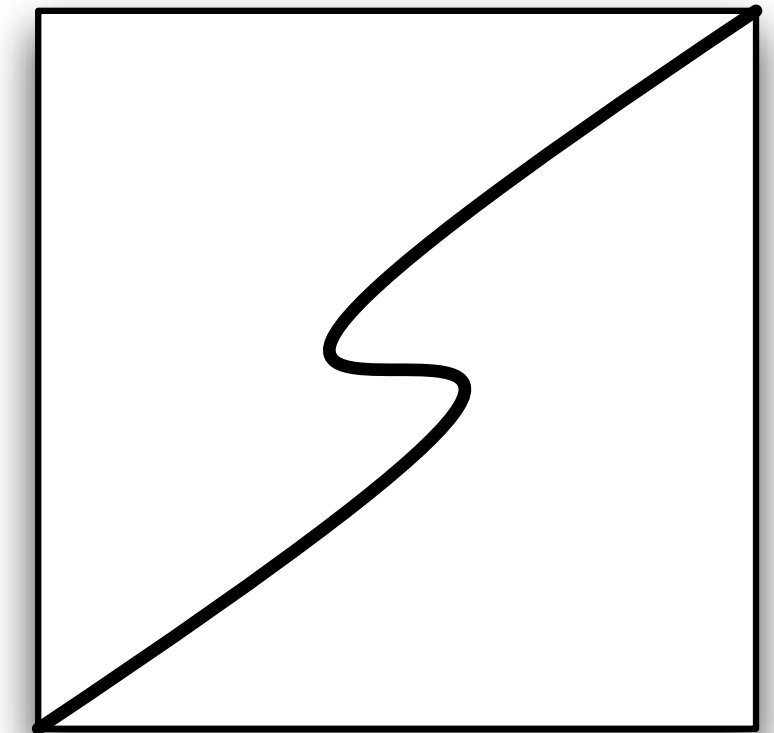
$(0; 0,5)$
 $(0,5; 1)$



$(1; 0,5)$
 $(0,5; 0)$



$(0; 2)$
 $(1; -1)$



$(1,5; 1)$
 $(-1,5; 0)$

Demo: Animation

Aktionen und Transaktionen

Implizite Animationen

- Layer animiert Änderungen automatisch
- Auslöser: Änderungen der Model-Layer
- Layer erzeugt `CAAction` über `actionForKey:`

Aktionserzeugung

- **Delegate:** `actionForLayer:forKey:`
- **Layer-Verzeichnisse:** `actions` und `style`
- **Layer:** `+defaultActionForKey:`

Animationserzeugung (I)

- `CAAnimation` implementiert `CAAction`
- Implementierung `runActionForKey:object:arguments:`
 - fügt Animation (`self`)
 - unter `key`
 - zum Layer (`object`) hinzu

Animationserzeugung (II)

- **Animationstyp:** `CABasicAnimation`
- **Startwert:** `fromValue` = aktueller Propertywert
- **Endwert:** `toValue` = `nil` (vom Layer)

Animationserzeugung (III)

```
- (id<CAAction>)actionForLayer:(CALayer *)inLayer  
  forKey:(NSString *)inKey {
```

```
}
```

Animationserzeugung (III)

```
- (id<CAAction>)actionForLayer:(CALayer *)inLayer
  forKey:(NSString *)inKey {
    if([inKey isEqualToString:@"depth"]) {

        else {
            return [super actionForLayer:inLayer forKey:inKey];
        }
    }
}
```

Animationserzeugung (III)

```
- (id<CAAction>)actionForLayer:(CALayer *)inLayer
  forKey:(NSString *)inKey {
    if([inKey isEqualToString:@"depth"]) {
      CABasicAnimation *theAnimation = [CABasicAnimation animation];

      theAnimation.fromValue = @(self.depth);
      theAnimation.duration = 2.0;
      return theAnimation;
    }
    else {
      return [super actionForLayer:inLayer forKey:inKey];
    }
  }
}
```

Parametrisierung?

```
- (id<CAAction>)actionForLayer:(CALayer *)inLayer
  forKey:(NSString *)inKey {
    if([inKey isEqualToString:@"depth"]) {
      CABasicAnimation *theAnimation = [CABasicAnimation animation];

      theAnimation.fromValue = @(self.depth);
      theAnimation.duration = 2.0;
      return theAnimation;
    }
    else {
      return [super actionForLayer:inLayer forKey:inKey];
    }
  }
}
```

Transaktionen

- kapseln alle Änderungen am Layer-Baum
- Parametrisieren Animationen
- *implizit* oder *explizit* über `CATransaction`

Transaktionsparameter

- **Dauer:** `+setAnimationDuration:`
- **Verlauf:** `+setAnimationTimingFunction:`
- **An / Aus:** `+setDisabledActions:`
- **beliebige Schlüssel-Wert-Paare:** `+setValueForKey:`

Explizite Transaktionen

```
[CATransaction begin];  
[CATransaction setAnimationDuration:2.0];  
[CATransaction setAnimationTimingFunction:  
    kCAMediaTimingFunctionEaseIn]  
theLayer.depth = 3.5;  
[CATransaction commit];
```

Animationserzeugung reloaded

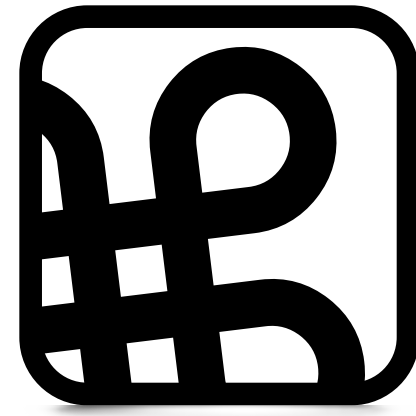
```
CABasicAnimation *theAnimation = [CABasicAnimation animation];  
  
theAnimation.fromValue = @(self.depth);  
theAnimation.duration = [CATransaction animationDuration];  
theAnimation.timingFunction =  
    [CATransaction animationTimingFunction];
```

Demo:

Explizite Transaktion

Fragen?

Vielen Dank



Macoun