

Praktikum Entwicklung von Mediensystemen mit iOS

WS 2011

Prof. Dr. Michael Rohs
michael.rohs@ifi.lmu.de
MHCI Lab, LMU München

Today

- Benutzer
- Hauptfunktionen
- 2 Interviews
- Storyboard

- Papier-Prototyp
- User-Test mit Papier-Prototyp
- Erfahrungen mit User-Test
- Start Software Prototyp

Timeline

#	Date	Topic
1	19.10.2011	Introduction and overview of iOS
2	26.10.2011	App architecture, touch input, saving data
3	2.11.2011	Location, networking, sensors
4	9.11.2011	iOS 5, storyboards, automatic reference counting
5	16.11.2011	Interviews, storyboarding; brainstorming
6	30.11.2011	Paper prototyping test, start of software prototype
7	14.12.2011	Heuristic evaluation of software prototype
8	11.1.2012	Think-aloud user study
9	25.1.2012	Completion of software prototype
10	1.2.2012	Final presentation

vorige Aufgabe: Initiierung des Projekts

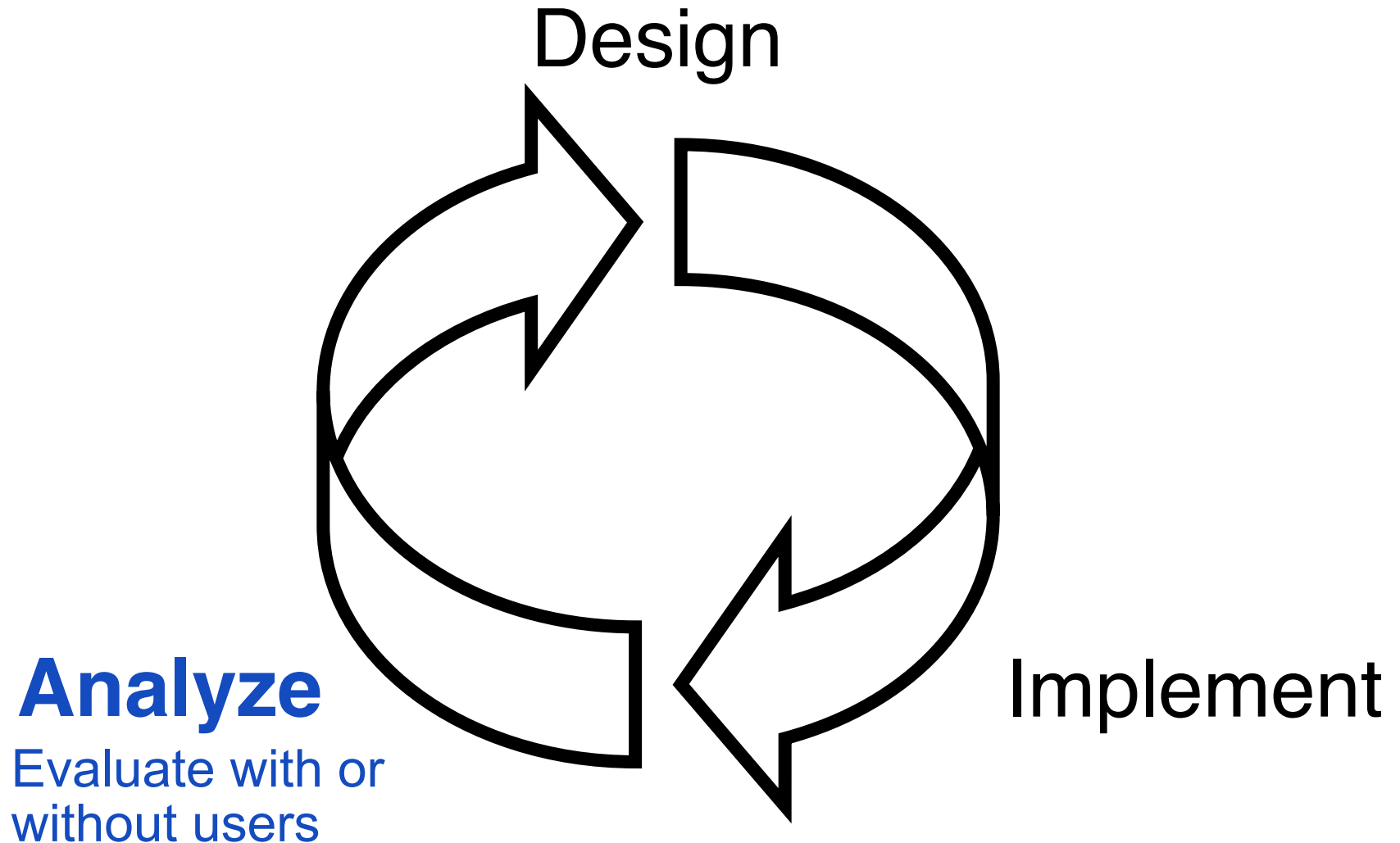
Dokumentieren Sie die Ergebnisse des Brainstormings (z.B. durch Einscannen der Ideenblätter). Einigen Sie sich innerhalb der Gruppe auf eine der drei vorgestellten Projektideen, die Sie im Praktikum umsetzen. Lösen Sie dann gemeinsam die folgenden Aufgaben:

- a) Beschreiben Sie knapp einen **typischen Benutzer** Ihrer App. Entwerfen Sie dazu eine „Persona“. Eine „Persona“ ist ein fiktiver aber typischer Nutzer, der die Zielgruppe repräsentiert.
- b) Überlegen Sie sich die **Hauptfunktionen** Ihrer App und notieren Sie weitere Features und Entwurfsideen. Diskutieren Sie gemeinsam, welche Funktionen am wichtigsten sind und beschränken Sie sich auf diese.
- c) Führen Sie **Interviews** mit 2 typischen Benutzern. Beschreiben Sie ihnen die Idee der App und finden Sie heraus, wie sie ankommt.
- d) Erstellen Sie ein **Storyboard** des Interaktionsablaufs.
- e) Erstellen Sie einen **Papier-Prototyp** der die einzelnen Screens und Abläufe darstellt.

→ Erstellen Sie eine PowerPoint/Keynote Präsentation zu diesen Punkten

EVALUATION

DIA Cycle: When to evaluate?



Heuristic Evaluation

- Choose usability heuristics
 - (general usability principles, e.g., Nielsen's 10 Usability Principles)
- Step through tasks and check whether guidelines are followed
- Severity rating for each problem (Nielsen)
 - 0 = I don't agree this is a problem at all
 - 1 = cosmetic problem
 - 2 = minor usability problem, low priority to fix
 - 3 = major usability problem, high priority to fix
 - 4 = usability catastrophe, imperative to fix before release
- + Quick and cheap
- Subjective (have several independent evaluators)
 - See also: www.useit.com/papers/heuristic

10 Usability Principles (Jakob Nielsen)

1. Keep the interface simple!
2. Speak the user's language!
3. Minimize the user's memory load!
4. Be consistent and predictable!
5. Provide feedback!
6. Design clear exits and closed dialogs!
7. Offer shortcuts for experts!
8. Help to recover from errors, offer Undo!
9. Prevent errors!
10. Include help and documentation!



Four Fundamental Concepts (Donald Norman)

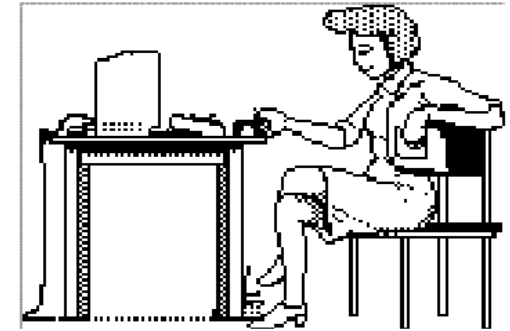


- Affordances & visibility
 - Affordances
 - http://www.jnd.org/dn.mss/affordances_and_design.html
 - Can the user tell the state of the system and the alternatives for action by looking at the system?
- Conceptual models
 - Is the user able to predict how actions affect the system?
- Natural mapping
 - Is it possible to determine the relationships between actions and results, between controls and effects?
- Feedback
 - Does the user receive full and continuous feedback about the results of actions?

User Interface Guidelines

- Concrete guidelines for look-and-feel and behavior
 - Visual appearance, e.g., icon design
 - Purpose of user interface elements
 - Layout of user interface elements
 - Behavior, conventions of system features
- Android User Interface Guidelines
 - http://developer.android.com/guide/practices/ui_guidelines/index.html
- iOS Human Interface Guidelines
 - <http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf>
 - Aesthetic integrity, consistency, direct manipulation, feedback, metaphors, user control, ...

Silent Observation



Source: Saul Greenberg

- Designer watches user in lab or in natural environment while working on one of the tasks
- No communication during observation
- + Helps discover big problems
- No understanding of decision process (that may be wrong) or user's mental model, opinions, or feelings

Think Aloud



Source: Saul Greenberg

- As Silent Observation, but user is asked to say aloud
 - What he thinks is happening (state)
 - What he is trying to achieve (goals)
 - Why he is doing something specific (actions)
- Most common method in industry
- + Good to get some insight into user's thinking, but:
 - Talking is hard while focusing on a task
 - Feels weird for most users to talk aloud
 - Conscious talking can change behavior

Evaluation in the Mobile Context

- Context of use needs to be taken into account
 - Factors: User, activity, device, environment
- Usage “on the move”
 - Physically moving: walking, driving a car, traveling as a passenger
 - Being in different places: away from office environment or home
- Difficult to collect data in the field
 - Recording data
 - Controlling experimental conditions



CORE ANIMATION LAYERS

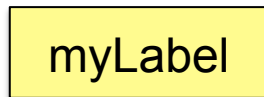
There is more to UIViews than meets the eye...

- UIViews contain CALayers
 - Normally not a problem, but may be when animating
- UI elements not responsive during animation
 - (Partial) solution: `UIViewAnimationOptionAllowUserInteraction`
- Example:

```
[UIView animateWithDuration:5.0 delay:0.0
    options:UIViewAnimationOptionAllowUserInteraction |
            UIViewAnimationOptionCurveEaseInOut
    animations:^(
        label.transform = CGAffineTransformMakeTranslation(500, 0);
    )
    completion:nil];
```

UIViews contain CALayers

- Problem: At start of animation, UIView frame immediately gets end-of-animation state
 - Touch input related to end-of-animation state
- Example: translation 500 points in x direction in 5s
 - Result: Label appears to be non-responsive during animation



- Why?
 - UIViews contain one or more Core Animation Layers (CALayers)
 - CALayers represent a rectangular area on the screen
 - CALayers have animated properties
 - CALayers do not process user input
 - UIViews process user input → view.frame

Core Animation Layers (class CALayer)

- Represent rectangular area on the screen
- Have animated properties
- Do not process user input (UIViews do)
 - UIView processes user input if touch point within view.frame
- UIView contains one or more CALayers
 - `CALayer* l = label.layer;`
- Can form hierarchies (superlayer, sublayers)
- Define geometry (position, size, transform)
- Can have content (image, Core Graphics drawing, color)
- Appearance (shadow, rounded edges, insets, etc.)
- Framework QuartzCore, `#import "QuartzCore/CALayer.h"`

Example CALayer Hierarchy

```
self.view.layer.backgroundColor =  
    [UIColor greenColor].CGColor;  
self.view.layer.cornerRadius = 10.0;  
self.view.layer.frame =  
    CGRectInset(self.view.layer.frame, 10, 10);
```

```
CALayer *imageLayer = [CALayer layer];  
imageNamed:@"testimage.png"].CGImage;
```

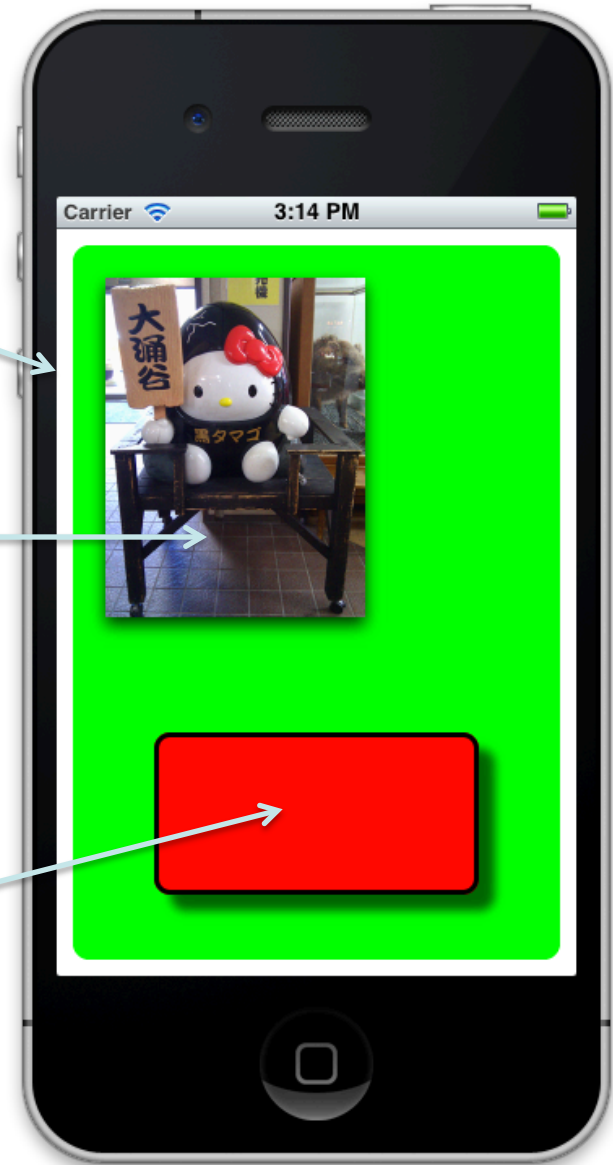
...

```
[self.view.layer addSublayer:imageLayer];
```

```
CALayer *sublayer = [CALayer layer];  
sublayer.backgroundColor =  
    [UIColor redColor].CGColor;
```

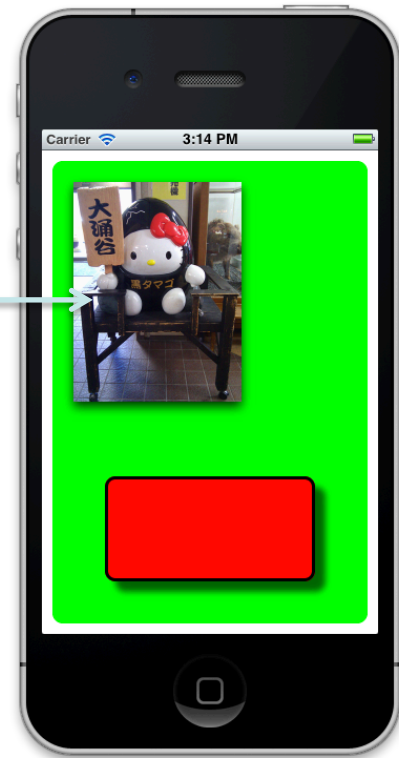
...

```
[self.view.layer addSublayer:sublayer];
```



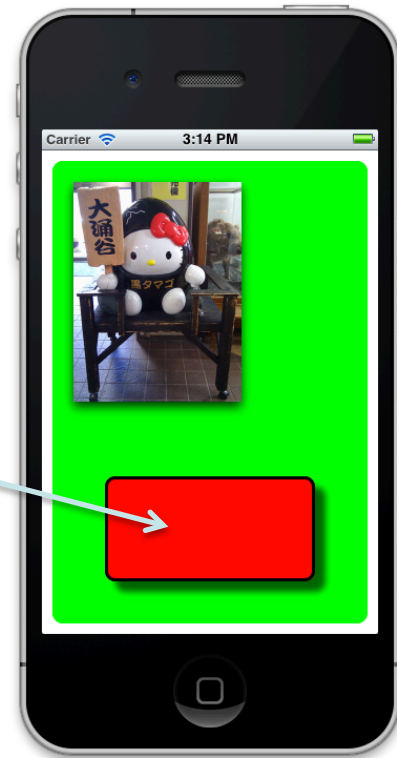
Example CALayer Hierarchy

```
CALayer *imageLayer = [CALayer layer];  
imageLayer.frame = CGRectMake(20, 20, 160, 214);  
imageLayer.cornerRadius = 10.0;  
imageLayer.contents = (id) [UIImage  
imageNamed:@"testimage.png"].CGImage;  
imageLayer.shadowOffset = CGSizeMake(0, 5);  
imageLayer.shadowRadius = 5.0;  
imageLayer.shadowColor = [UIColor blackColor].CGColor;  
imageLayer.shadowOpacity = 0.8;  
[self.view.layer addSublayer:imageLayer];
```



Example CALayer Hierarchy

```
CALayer *sublayer = [CALayer layer];  
sublayer.backgroundColor = [UIColor redColor].CGColor;  
sublayer.shadowOffset = CGSizeMake(10, 10);  
sublayer.shadowRadius = 3.0;  
sublayer.shadowColor = [UIColor blackColor].CGColor;  
sublayer.shadowOpacity = 0.6;  
sublayer.frame = CGRectMake(50, 300, 200, 100);  
sublayer.borderColor = [UIColor blackColor].CGColor;  
sublayer.borderWidth = 3.0;  
sublayer.cornerRadius = 10.0;  
[self.view.layer addSublayer:sublayer];
```



Core Animation Rendering Architecture

- CALayers encapsulate geometry, timing, visual properties
- Visible layer tree is backed by presentation and render trees
 - Layer tree: property values (model)
 - Presentation tree: state that is currently shown, gradually changes during animation

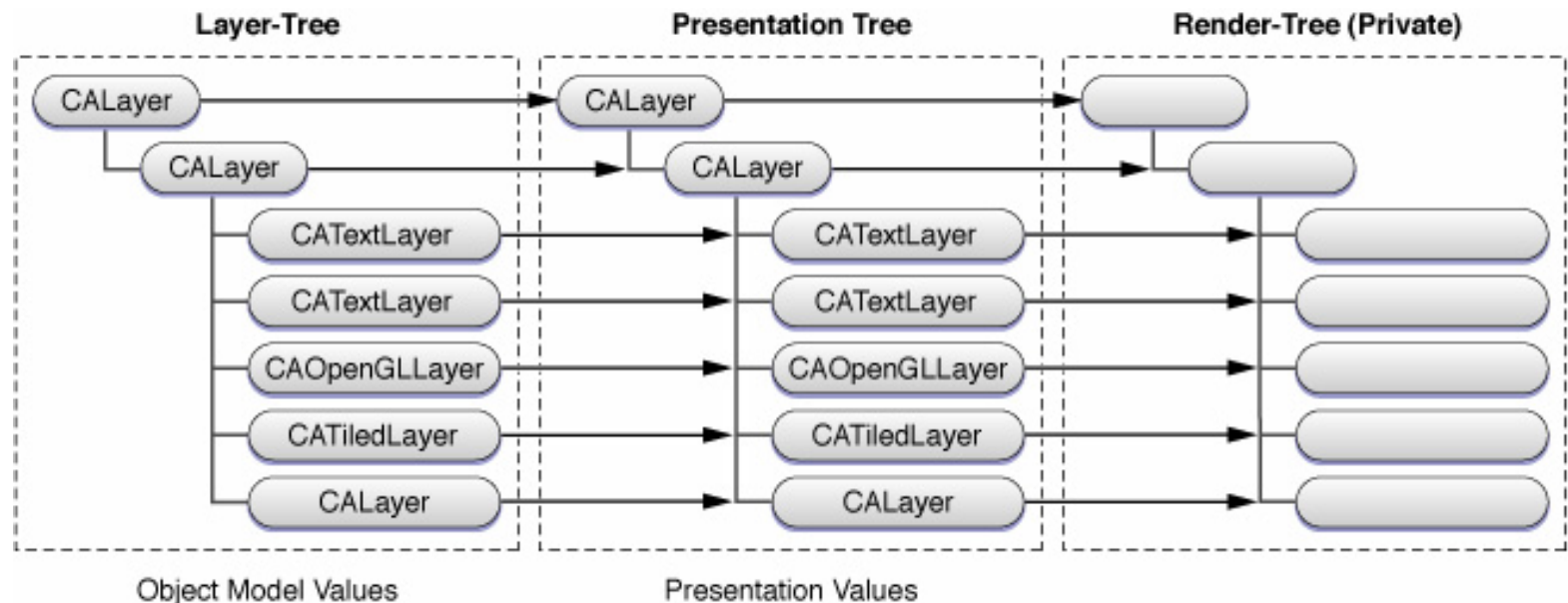


Image source: Apple documentation

CALayer Geometry and Transforms

- Each layer has its own coordinate system
 - Position, frame, bounds, anchor point
- Transform as known from UIViews

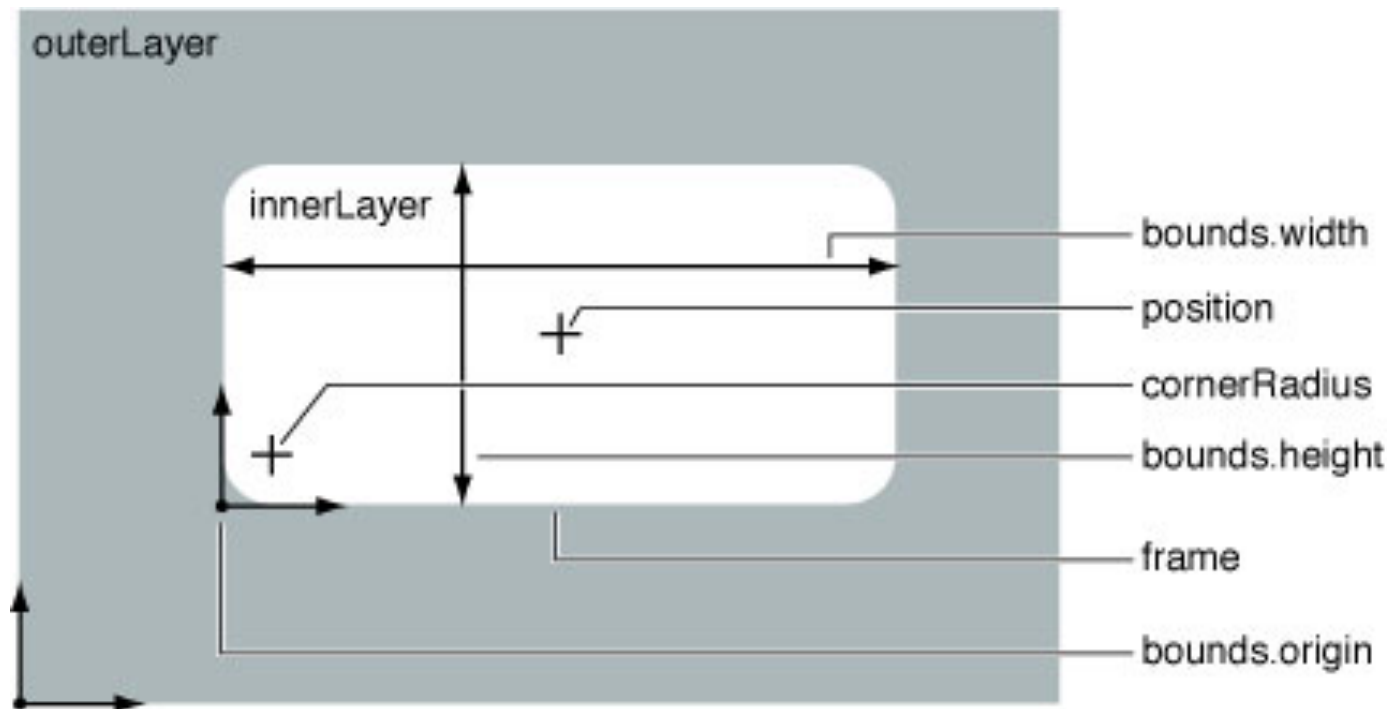


Image source: Apple documentation

Resources on Core Animation

- http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/CoreAnimation_guide
- <http://www.raywenderlich.com/2502/introduction-to-calayers-tutorial>
- <http://www.raywenderlich.com/2454/how-to-use-uiview-animation-tutorial>

Solution: Derive from UIView-Subclass

- Implement hit test your self

```
- (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event {
    UIView *v = [super hitTest:point withEvent:event];
    return v;
}

- (BOOL)pointInside:(CGPoint)point withEvent:(UIEvent *)event {
    CALayer *modelLayer = [[self layer] modelLayer];
    CALayer *presentationLayer = [[self layer] presentationLayer];
    CGPoint point2 = [presentationLayer convertPoint:point
                    fromLayer:modelLayer];
    BOOL b = [presentationLayer containsPoint:point2];
    return b;
}
```


AUFGABE

Aufgabe

- Erstellen Sie einen Papier-Prototyp, der die einzelnen Screens/UI-Zustände und Übergänge dazwischen darstellt.
- Führen Sie einen Benutzer-Test mit dem Papier-Prototyp mit 3 Benutzern (nicht aus dem PEM-Praktikum) durch
 - Typische Aufgabe festlegen
 - Benutzer bitten, diese Aufgabe mit dem Papier-Prototyp durchzuführen
 - Protokollieren, Probleme finden
 - Papier-Prototyp verbessern
- Schreiben Sie die Erfahrungen beim Test auf
 - Was ließ sich gut testen, was war problematisch
- Starten Sie den Software Prototyp