

Views, basic principles

Fabrice.Kordon@lip6.fr




As an introduction...


 **View = canonical element in iOS (and Android)**

 Includes interaction mechanisms


 **What is it?**

 A rectangle

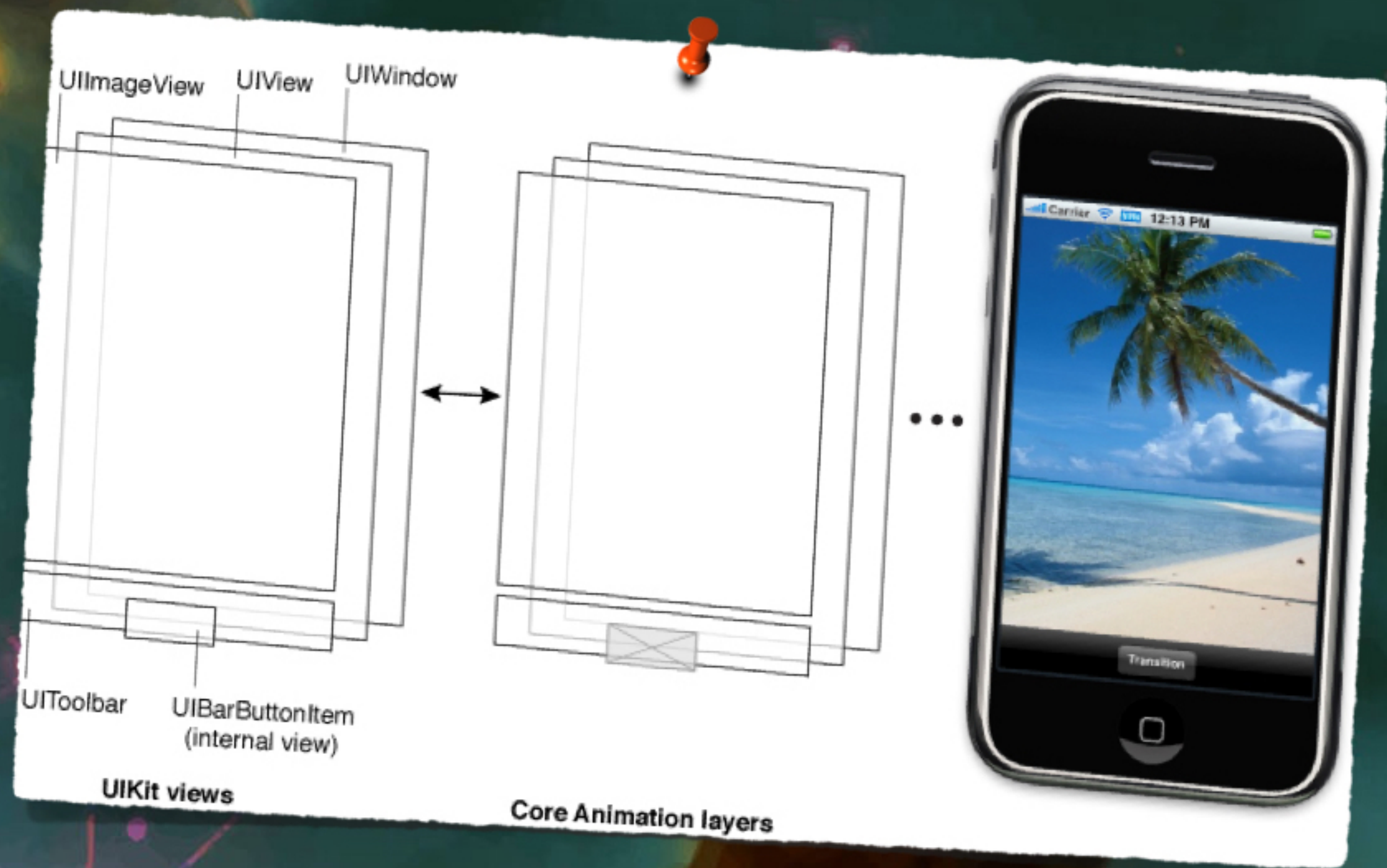
- ▶ To draw
- ▶ To insert subviews
- ▶ Sensible to events

 Hierarchical organization

- ▶ 1 super view
- ▶ 0+ subview

 Recover what's behind

- ▶ Opacity




As an introduction...


 **View = canonical element in iOS (and Android)**

 Includes interaction mechanisms


 **What is it?**

 A rectangle

- ▶ To draw
- ▶ To insert subview
- ▶ Sensible to events

 Hierarchical organization

- ▶ 1 super view
- ▶ 0+ subview

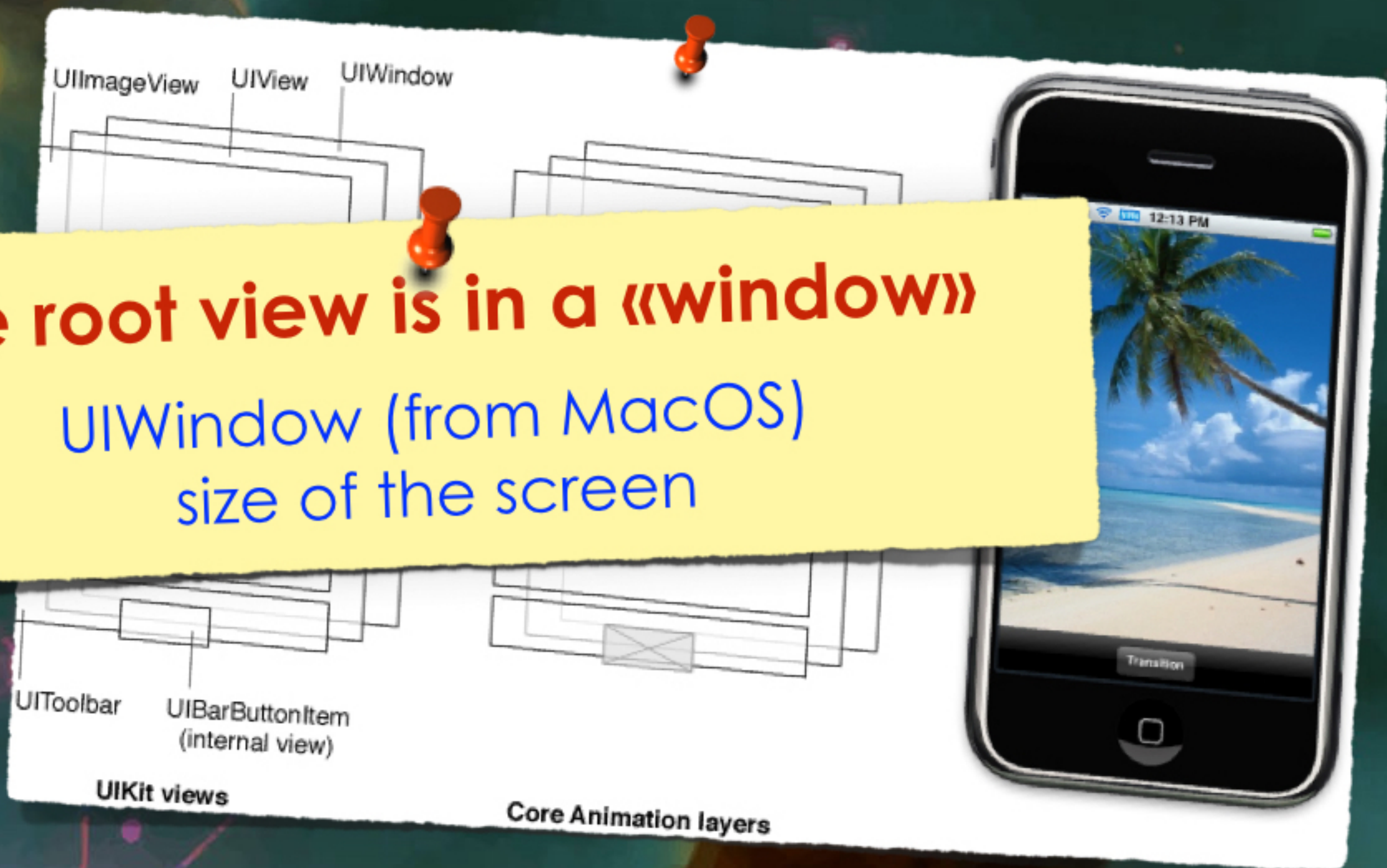
 Recover what's behind

- ▶ Opacity



The root view is in a «window»

UIWindow (from MacOS)
size of the screen



UIView, main features

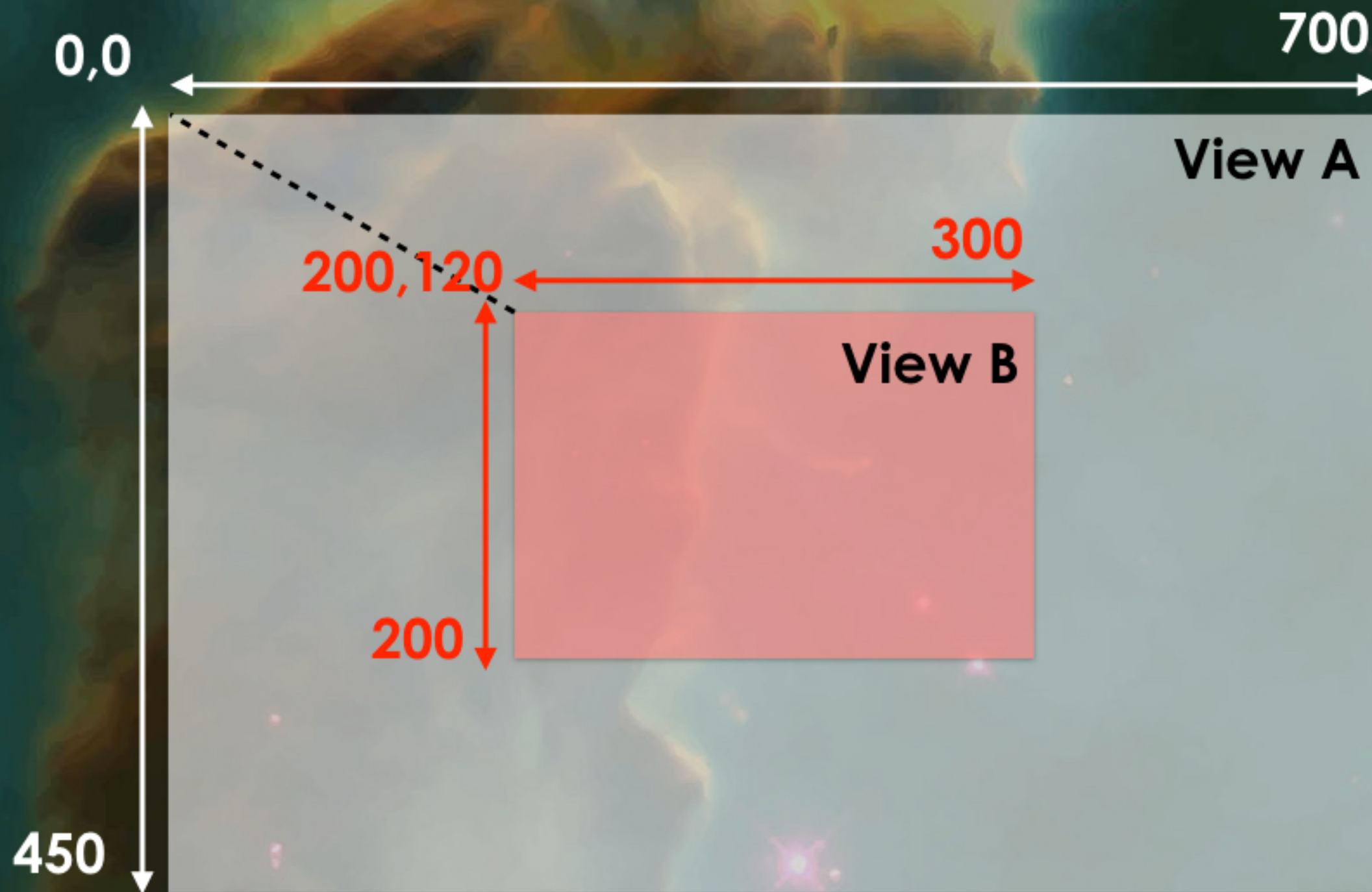
Customization

- frame position + coordinate in parent view
- bounds origin + size
- backgroundColor background color (default = clearColor)
- hidden boolean (false if view is visible)
- etc.

relationships with other views

- superview reference to the super view
- subview array of contained views
- windows the window containing the view (or nil)
- etc.

View & coordinates



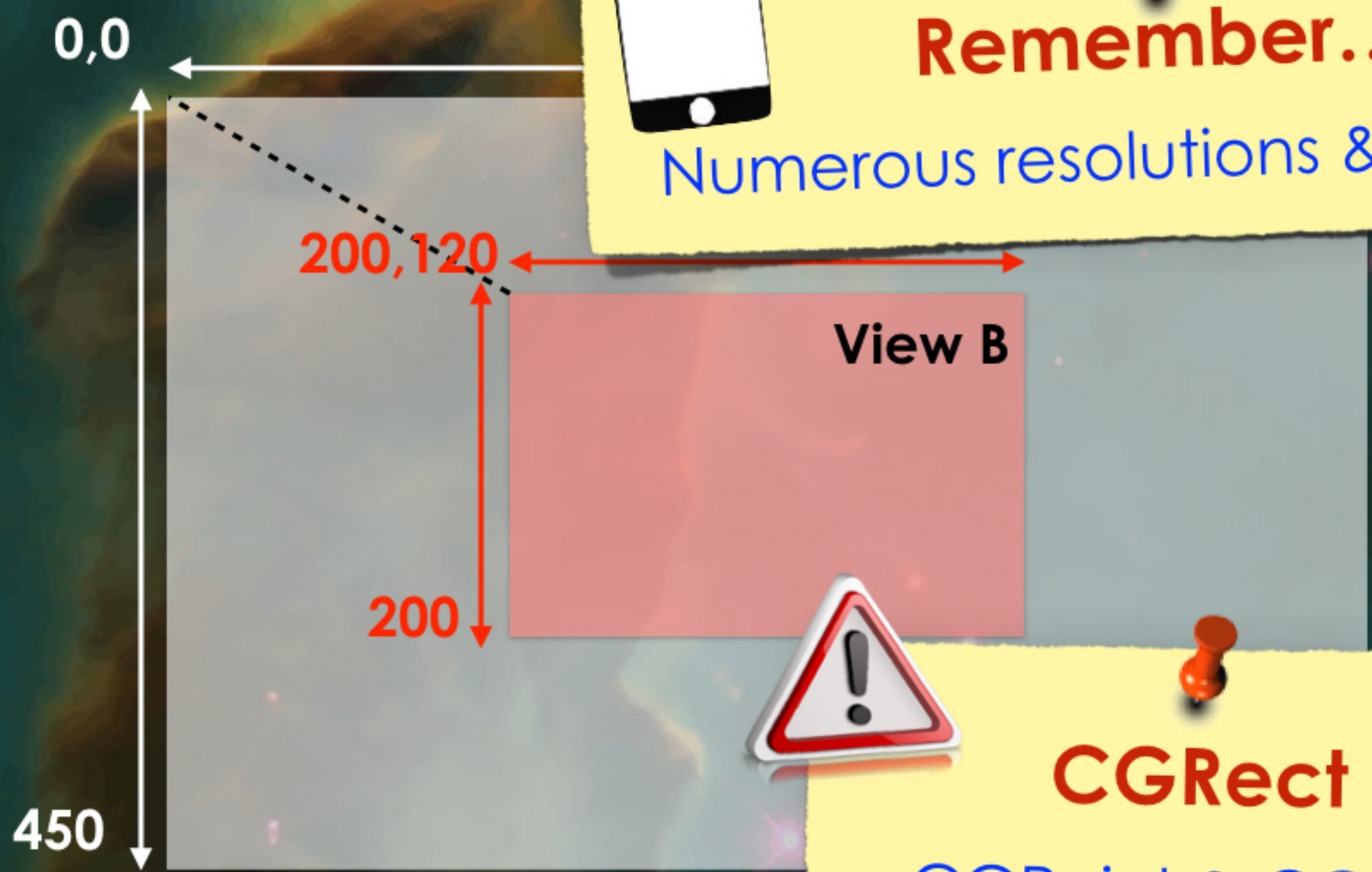
- ▶ **View A's frame**
origin : 0,0
size : 700,450
- ▶ **View A's bound**
origin : 0,0
size : 700,320
- ▶ **View B's frame**
origin : 200,120
size : 300,200
- ▶ **View B's bound**
origin : 0,0
size : 300,200

- Frame = configuration seen from «outside»
- Bounds = configuration seen from «inside»

View & coordinates



Remember...
Numerous resolutions & densities



- ▶ View A's frame
origin : 0,0
size : 700,450
- ▶ View A's bound
origin : 0,0
size : 700,320
- ▶ View B's frame
origin : 200,120
size : 300,200
- ▶ View B's bound
origin : 0,0
size : 300,200



CGRect
CGPoint & CGSize

- Frame = configuration seen from «outside»
- Bounds = configuration seen from «inside»

Manipulating views



Objective-C

- (void)addSubview:(UIView*) view;
- (void)removeFromSuperview;
- (void)insertSubview:(UIView*)view atIndex:(int)index;
- (void)exchangeSubviewAtIndex:(int)index
withSubviewAtIndex:(int)otherIndex;



Swift

```
func addSubview(_ view: UIView)

func insertSubview(view: UIView,
                  aboveSubview: UIView)

func exchangeSubviewAtIndex(at: Int,
                            withSubviewAt: Int)

func removeFromSuperview()
```

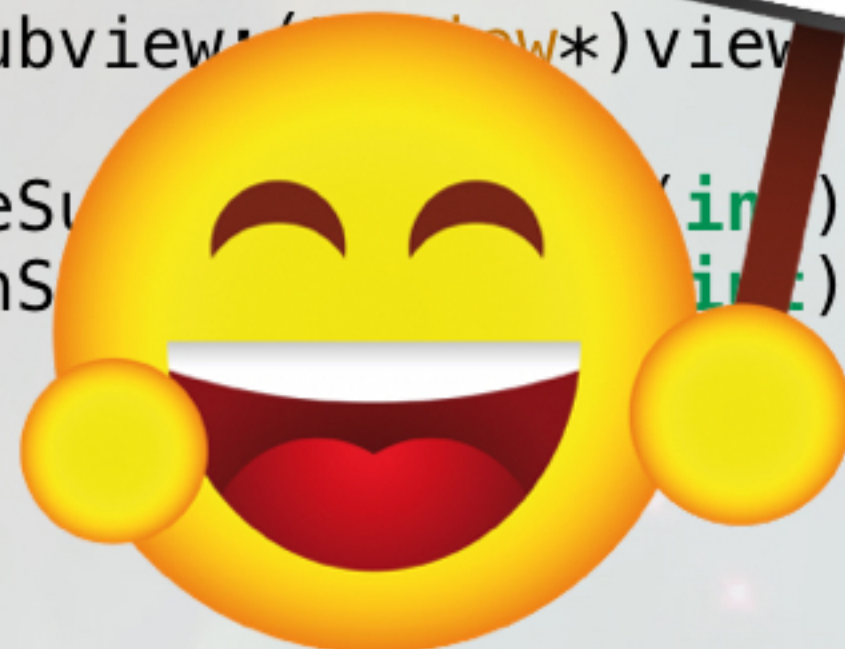
Manipulating views



Objective-C

- (void) addSubview: (UIView*)
- (void) removeFromSuperview
- (void) insertSubview:(UIView*)view atIndex:(int) index;
- (void) exchangeSubviewAtIndex:(int) index withSubviewAtIndex:(int) otherIndex;

THERE ARE MANY MORE!



Swift

- func addSubview(_ view: UIView)
- func insertSubview(view: UIView, aboveSubview: UIView)
- func exchangeSubviewAtIndex(at: Int, withSubviewAtIndex: Int)
- func removeFromSuperview()

RTFM!



Create your view



Process

- 📱 Create a class that inherits from UIView
- 📱 Implement the body of this class
 - ▶ **Dedicated methods**
- 📱 You may associate it to a controller



Important methods to implement

- 📱 Display the view
 - ▶ **Essentially for graphical views**
 - `override func draw(_ rect: CGRect)`
 - `(void)drawRect:(CGRect)rect`
- 📱 Enforce view «refresh»
 - ▶ **setNeedsDisplay**

As a conclusion

All is view!!!!

- You must install object or draw in views

With UIKit

- Only rectangles
 - ▶ **Transparence is useful (e.g. icons)**
- Numerous mechanisms available
 - ▶ **UILabel, UIButton, UISlider, UIStepper, etc.**

With CoreGraphics

- A full library for drawing
- Definition of a context (draw color, fill color, etc.)
- Forms («path» to draw)
- Font,
- etc.

As a conclusion

All is view!!!!

- You must install object-oriented programming



Other mechanisms?

OpenGL (ES), Metal, SceneKit (3D)

With UIKit

- Only rectangles
 - ▶ Transparency is useful (e.g. icons)
- Numerous mechanisms available
 - ▶ UILabel, UIButton, UISlider, UIStepper, etc.



Sorry, not discussed

Would be too long...



With CoreGraphics

- A full library for drawing
- Definition of a path
- Forms («path» to draw)
- Font,
- etc.

RTFM

