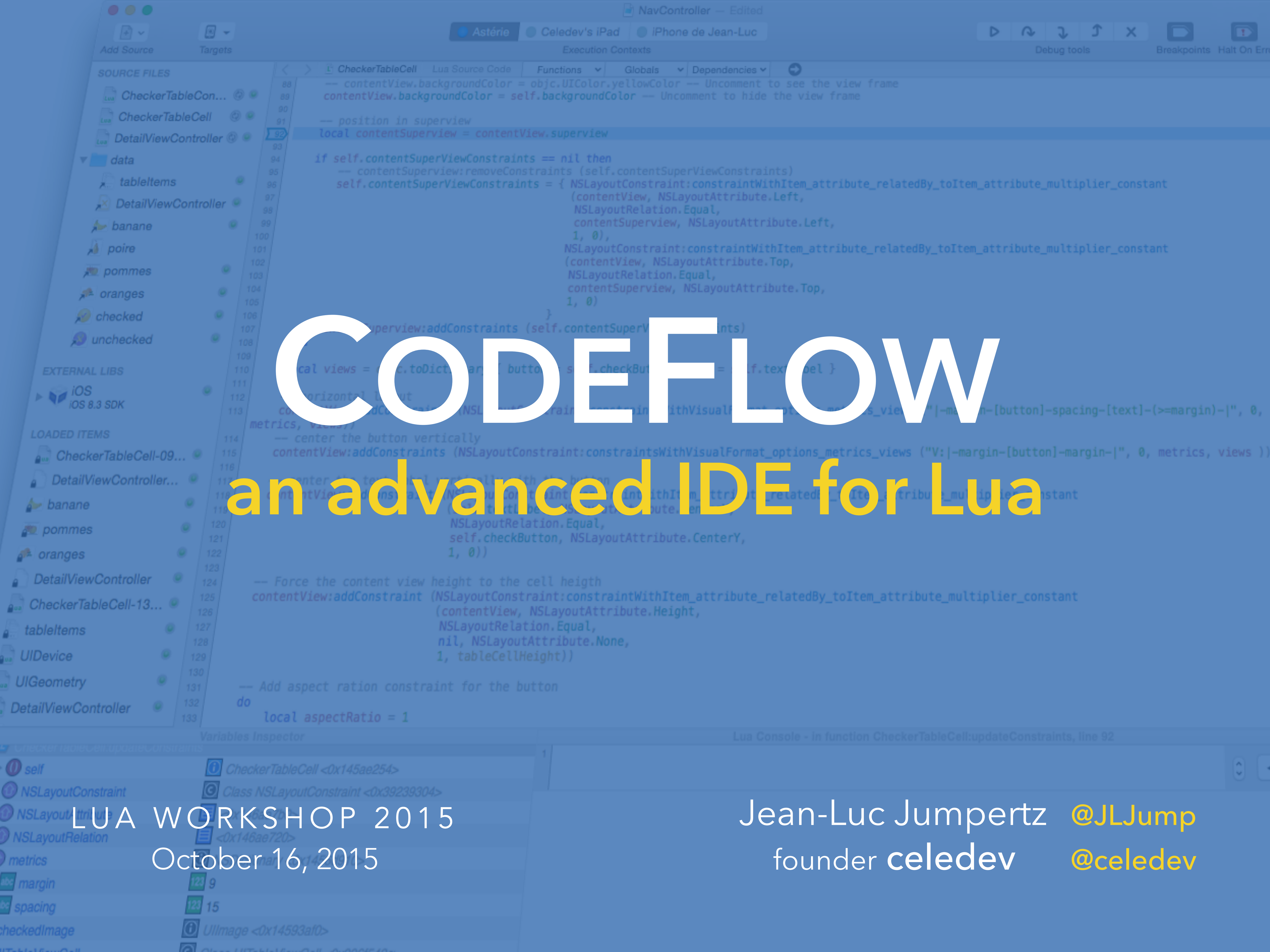


```
    -- contentView.backgroundColor = objc.UIColor.yellowColor -- Uncomment to see the view frame
    contentView.backgroundColor = self.backgroundColor -- Uncomment to hide the view frame

    -- position in superview
    local contentSuperview = contentView.superview

    if self.contentSuperViewConstraints == nil then
        -- contentSuperview:removeConstraints (self.contentSuperViewConstraints)
        self.contentSuperViewConstraints = { NSLayoutConstraint:constraintWithItem_attribute_relatedBy_toItem_attribute_multiplier_constant
                                              (contentView, NSLayoutAttribute.Left,
                                               NSLayoutRelation.Equal,
                                               contentSuperview, NSLayoutAttribute.Left,
                                               1, 0),
                                             NSLayoutConstraint:constraintWithItem_attribute_relatedBy_toItem_attribute_multiplier_constant
                                             (contentView, NSLayoutAttribute.Top,
                                              NSLayoutRelation.Equal,
                                              contentSuperview, NSLayoutAttribute.Top,
                                              1, 0)
                                           }
          uperview:addConstraints (self.contentSuperV     ints)

     cal views =   c.toDict   ar   butto   s   f.checkBut      =   f.text   bel }

       orizont    l   ut
    co              dCo     rai   (NSL          rai      nstrain     ithVisualF     t   opt     me      s_vie    "|-ma    n-[button]-spacing-[text]-(>=margin)-|", 0,
metrics, views))
    -- center the button vertically
    contentView:addConstraints (NSLayoutConstraint:constraintsWithVisualFormat_options_metrics_views ("V:|-margin-[button]-margin-|", 0, metrics, views ))

                    NSLayoutRelation.Equal,
                    self.checkButton, NSLayoutAttribute.CenterY,
                    1, 0))

    -- Force the content view height to the cell heigth
    contentView:addConstraint (NSLayoutConstraint:constraintWithItem_attribute_relatedBy_toItem_attribute_multiplier_constant
                               (contentView, NSLayoutAttribute.Height,
                                NSLayoutRelation.Equal,
                                nil, NSLayoutAttribute.None,
                                1, tableCellHeight))

    -- Add aspect ration constraint for the button
    do
        local aspectRatio = 1
```

# CodeFlow

## an advanced IDE for Lua

LUA WORKSHOP 2015
October 16, 2015

Jean-Luc Jumpertz @JLJump
founder celedev @celedev

# CodeFlow

## Interactive Development of iOS / OS X Apps in Lua

Instant feedback

on real devices

full SDK scope

powerful dev tools

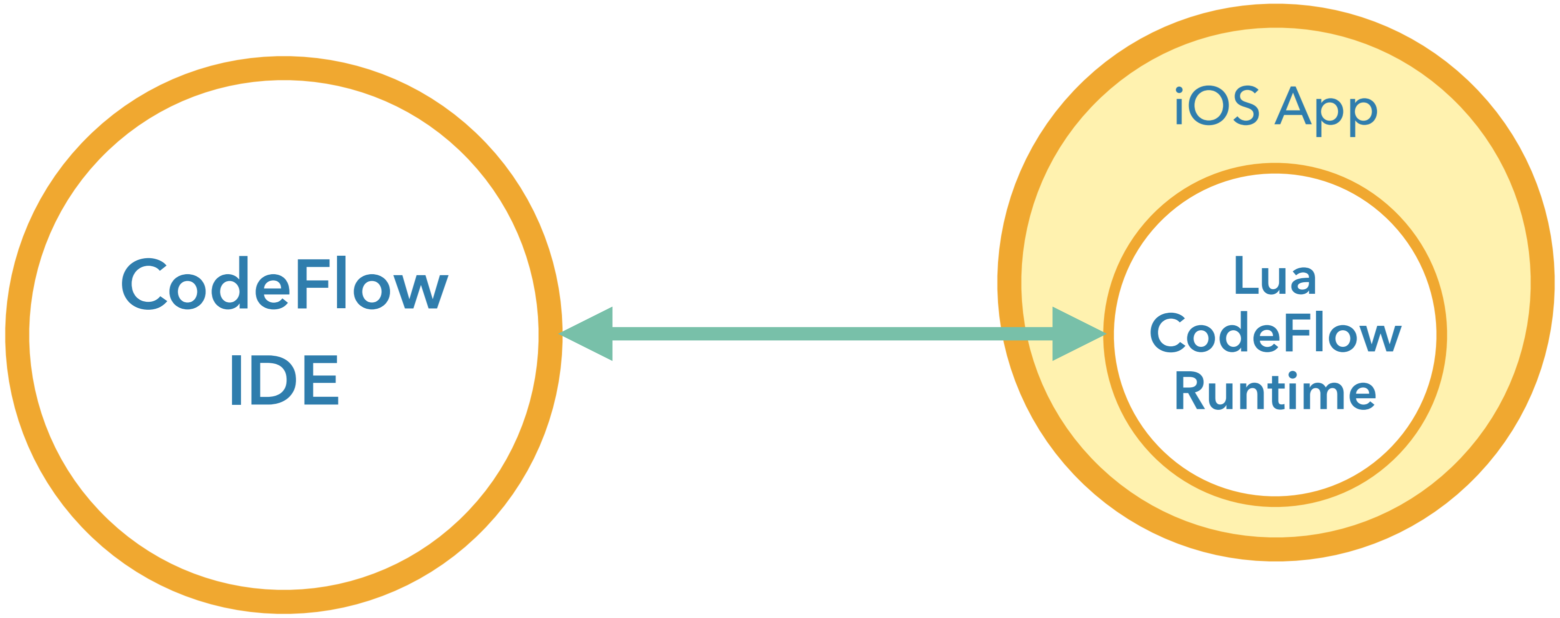
⇒ Faster development cycle

⇒ Easier experimentation, higher creativity

⇒ Better App UX

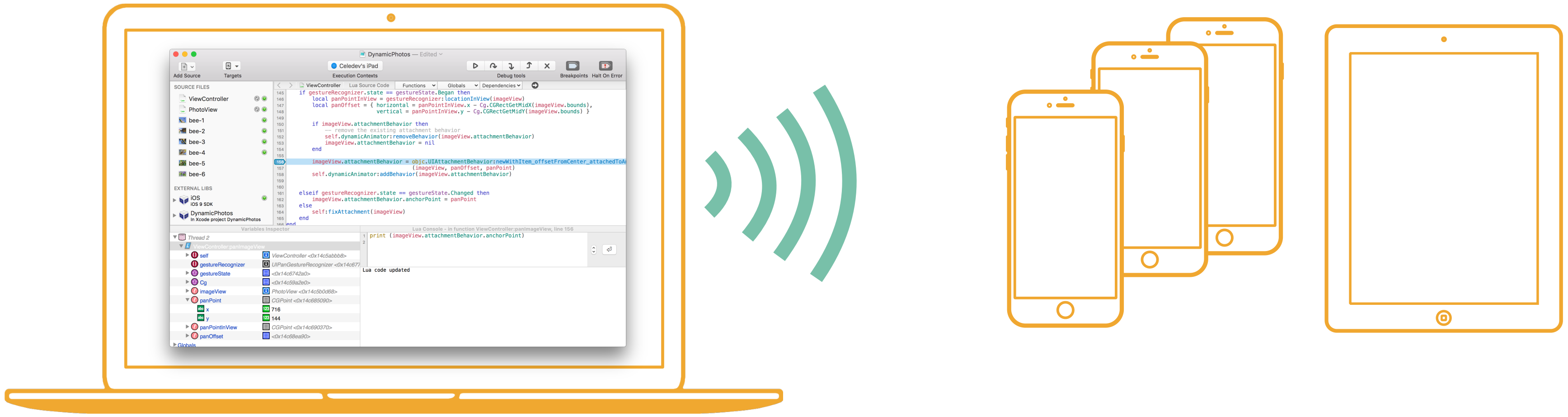
# CodeFlow

## Interactive Development of iOS / OS X Apps in Lua



# CodeFlow

## Interactive Development of iOS / OS X Apps in Lua



```
    if gestureRecognizer.state == gestureState.Began then
        local panPointInView = gestureRecognizer:locationInView(imageView)
        local panOffset = { horizontal = panPointInView.x - Cg.CGRectGetMidX(imageView.bounds),
                            vertical = panPointInView.y - Cg.CGRectGetMidY(imageView.bounds) }

        if imageView.attachmentBehavior then
            -- remove the existing attachment behavior
            self.dynamicAnimator:removeBehavior(imageView.attachmentBehavior)
            imageView.attachmentBehavior = nil
        end

        imageView.attachmentBehavior = objc.UIAttachmentBehavior:newWithItem_offsetFromCenter_attachedToA
                                           (imageView, panOffset, panPoint)
        self.dynamicAnimator:addBehavior(imageView.attachmentBehavior)


    elseif gestureRecognizer.state == gestureState.Changed then
        imageView.attachmentBehavior.anchorPoint = panPoint
    else
        self:fixAttachment(imageView)
    end
end
```

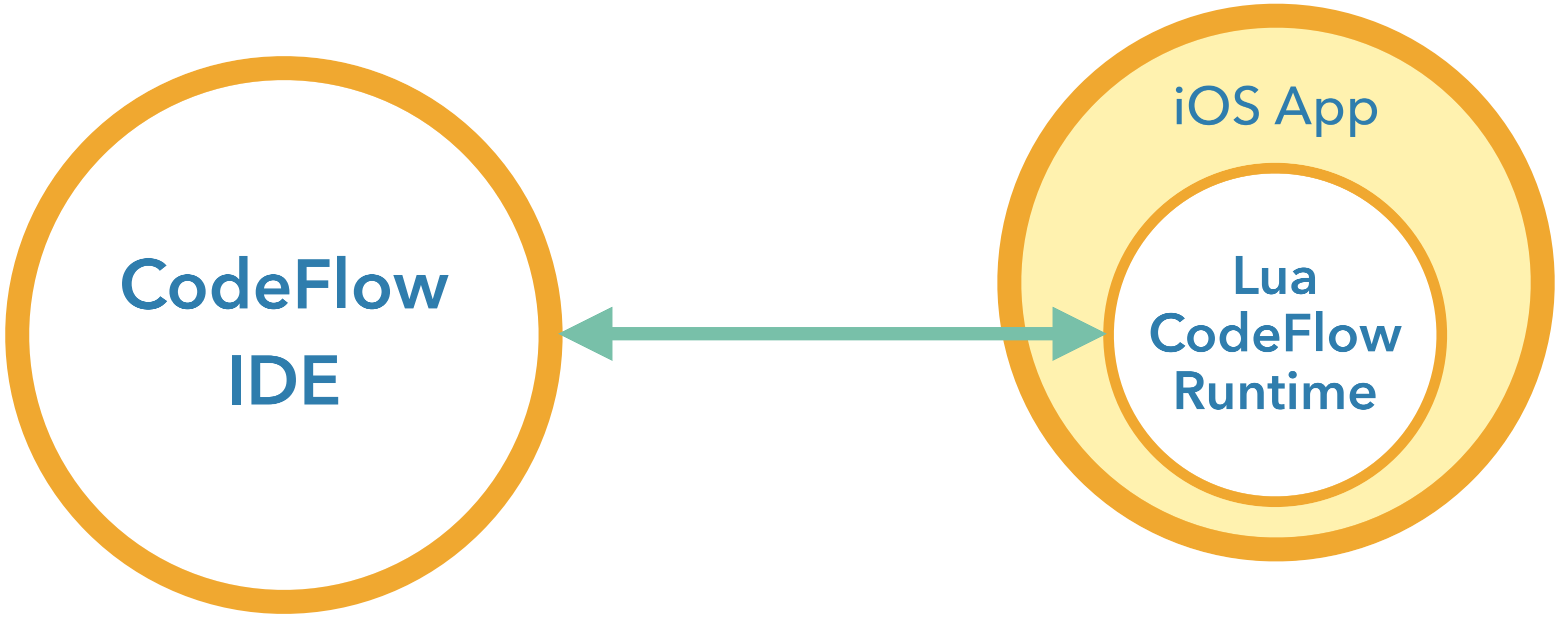
Thread 2

ViewController:panImageView

| | |
|---|---|
| self | ViewController <0x14c5abbb8> |
| gestureRecognizer | UIPanGestureRecognizer <0x14c677 |
| gestureState | <0x14c6742a0> |
| Cg | <0x14c59a2e0> |
| imageView | PhotoView <0x14c5b0d68> |
| panPoint | CGPoint <0x14c685090> |
| x | 716 |
| y | 144 |
| panPointInView | CGPoint <0x14c690370> |
| panOffset | <0x14c68ea90> |

```
print (imageView.attachmentBehavior.anchorPoint)
```

```
Lua code updated
```



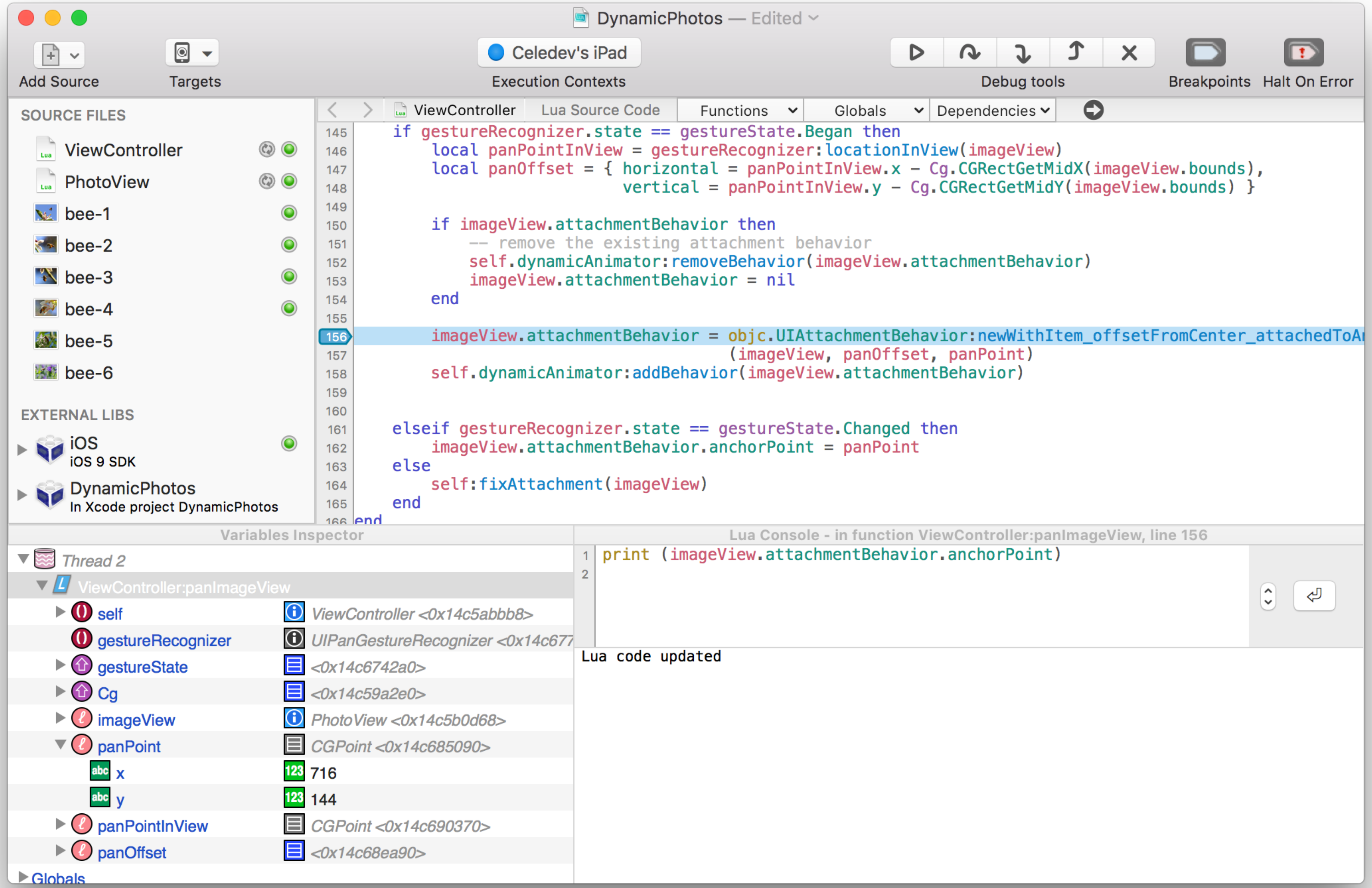
CodeFlow IDE

iOS App

Lua CodeFlow Runtime

# CodeFlow

## Interactive Development of iOS / OS X Apps in Lua



```
    if gestureRecognizer.state == gestureState.Began then
        local panPointInView = gestureRecognizer:locationInView(imageView)
        local panOffset = { horizontal = panPointInView.x - Cg.CGRectGetMidX(imageView.bounds),
                            vertical = panPointInView.y - Cg.CGRectGetMidY(imageView.bounds) }

        if imageView.attachmentBehavior then
            -- remove the existing attachment behavior
            self.dynamicAnimator:removeBehavior(imageView.attachmentBehavior)
            imageView.attachmentBehavior = nil
        end

        imageView.attachmentBehavior = objc.UIAttachmentBehavior:newWithItem_offsetFromCenter_attachedToA
                                      (imageView, panOffset, panPoint)
        self.dynamicAnimator:addBehavior(imageView.attachmentBehavior)


    elseif gestureRecognizer.state == gestureState.Changed then
        imageView.attachmentBehavior.anchorPoint = panPoint
    else
        self:fixAttachment(imageView)
    end
end
```

| | |
|---|---|
| Thread 2 | |
| ViewController:panImageView | |
| self | ViewController <0x14c5abbb8> |
| gestureRecognizer | UIPanGestureRecognizer <0x14c677 |
| gestureState | <0x14c6742a0> |
| Cg | <0x14c59a2e0> |
| imageView | PhotoView <0x14c5b0d68> |
| panPoint | CGPoint <0x14c685090> |
| x | 716 |
| y | 144 |
| panPointInView | CGPoint <0x14c690370> |
| panOffset | <0x14c68ea90> |
| Globals | |

```
print (imageView.attachmentBehavior.anchorPoint)
```

```
Lua code updated
```

# Lua code editor

- Code editor common features: code coloring, indentation, completion…
- **Standard editors** manage code purely as text

Lua mode

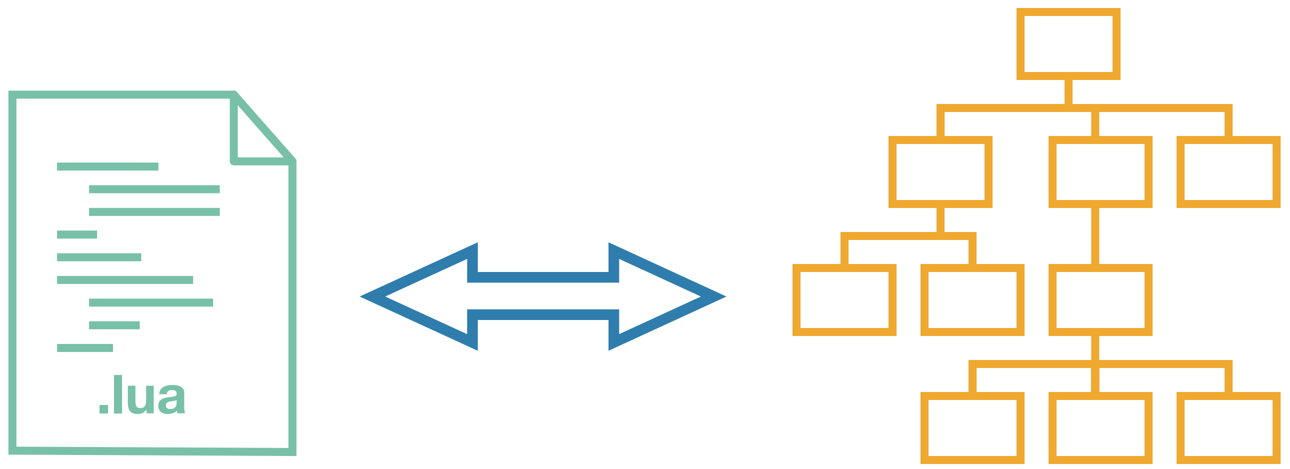
xml, json

- 1 file / language
- regex, list of keywords…

➡ The editor has no real knowledge of the program structure!

➡ Limited features set

# Syntax-driven Lua code editor



Source code

Lua AST

- The Lua Abstract Syntax Tree represents the syntax of a given Lua source file
- Each AST node stores the associated char range in the source file
- The AST must be kept in sync with the source code (asynchronously)

➡ All syntax-related actions in the editor use the corresponding AST

➡ Much more powerful and accurate than text-based processing

# Syntax-driven Lua code editor

- Demo
  - ➡ An advanced syntax-aware Lua editor can be a true help for the developer!

- Under the hood: building the AST
  - Many possible strategies
  - In CodeFlow, Lua AST generation is based on the standard Lua parser (`llex.c, lparser.c`)
    - ➡ very good performance
    - ➡ same language understanding as the Lua runtime
  - Source available at bitbucket.org/jean_luc/luasyntaxer

# Lua debugger

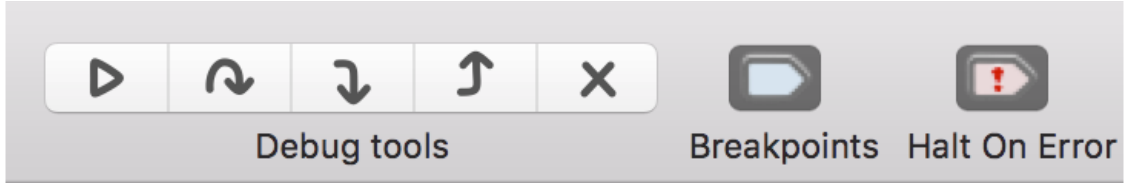
- For many developers, prints in the console is still the primary debug tool. 😟

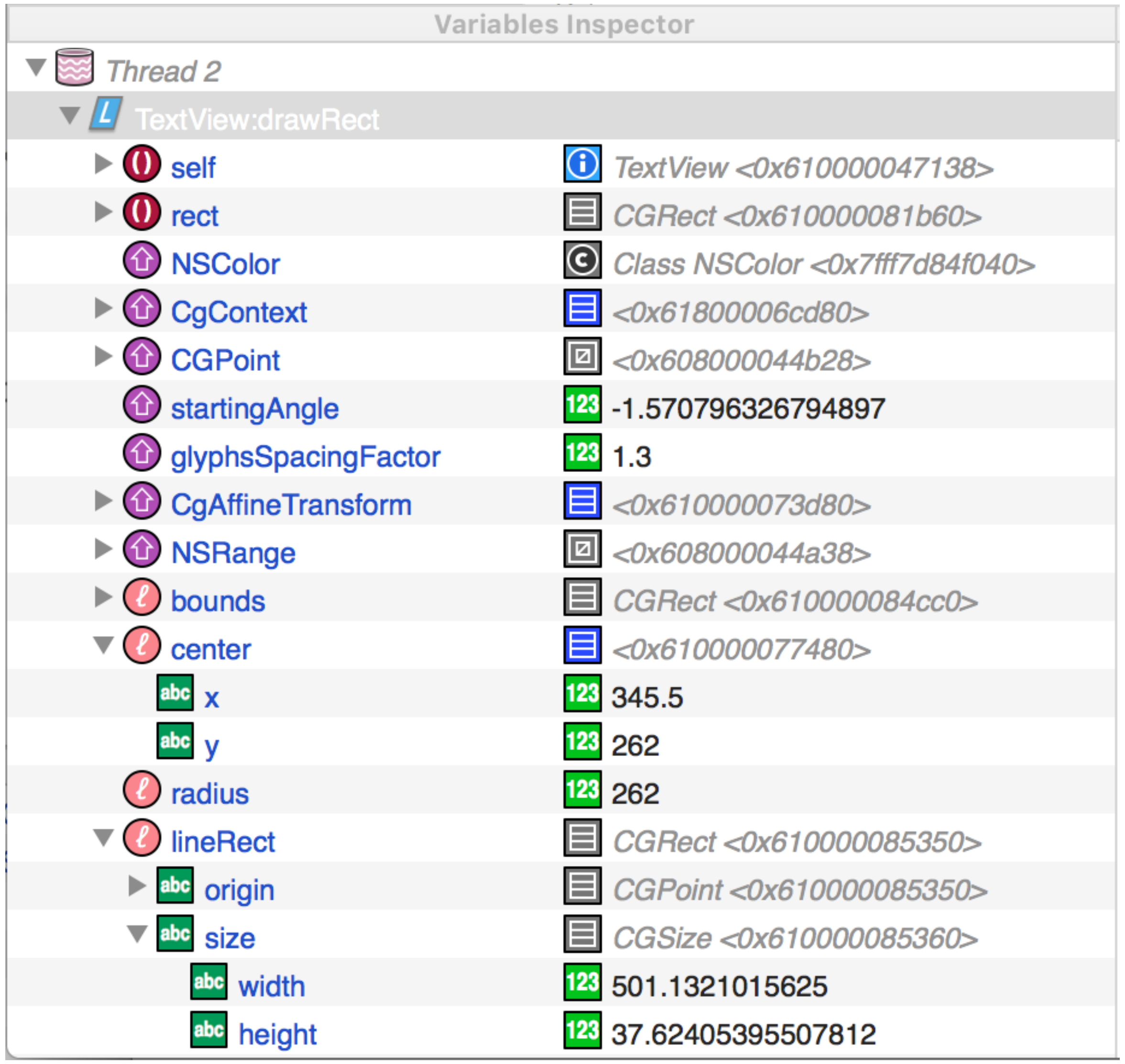
- **A good debugger makes bug fixing much easier!**
- The ideal debugger should be…
  - ➡ Easy to use and well-integrated with the rest of the IDE
  - ➡ Fast, low overhead
  - ➡ Transparent to use

    i.e not forcing you to change your program code to make it debug-compliant

# Debugger features

- step-by-step execution and breakpoints
- runtime errors analysis (+ make them recoverable)
- variable inspector / editor
- stack-context-aware command console
- multi-thread debugging



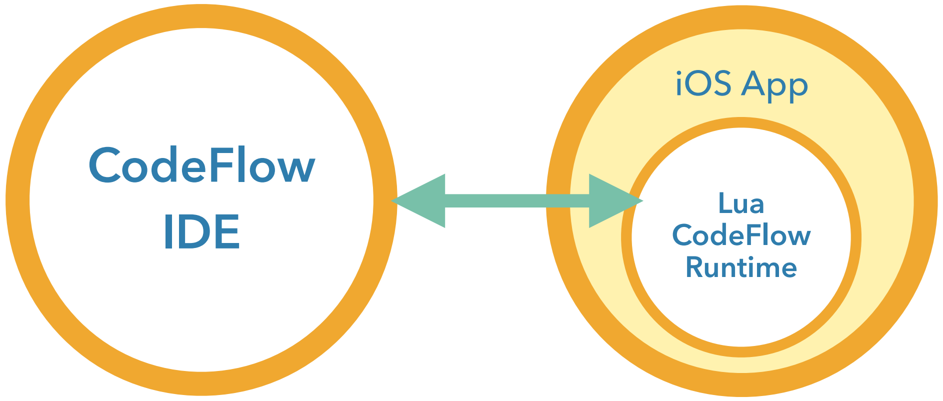


Variables Inspector

```
▼ Thread 2
  ▼ TextView:drawRect
    ▶ self                 TextView <0x610000047138>
    ▶ rect                 CGRect <0x610000081b60>
      NSColor              Class NSColor <0x7fff7d84f040>
    ▶ CgContext            <0x61800006cd80>
    ▶ CGPoint              <0x608000044b28>
      startingAngle        -1.570796326794897
      glyphsSpacingFactor  1.3
    ▶ CgAffineTransform    <0x610000073d80>
    ▶ NSRange              <0x608000044a38>
    ▶ bounds               CGRect <0x610000084cc0>
    ▼ center               <0x610000077480>
        x                  345.5
        y                  262
      radius               262
    ▼ lineRect             CGRect <0x610000085350>
      ▶ origin             CGPoint <0x610000085350>
      ▼ size               CGSize <0x610000085360>
          width            501.1321015625
          height           37.62405395507812
```

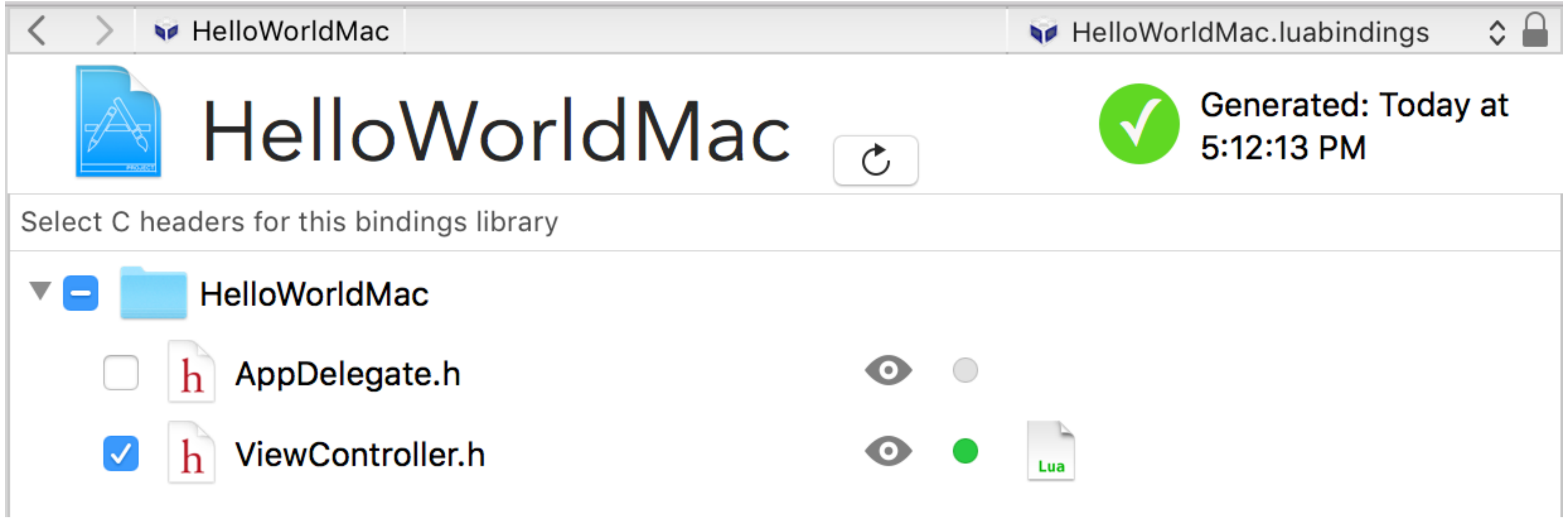
# Lua Modules in CodeFlow

- Managing modules is a classic role of an IDE
- A logical choice for Lua modules interaction: use the **`require`** function
- In CodeFlow, Lua modules are in the IDE space; but **`require`** is called from the app on the device
  - ➡ Custom **`require`** function in the CodeFlow Runtime
- Behavior changes
  - Enable multiple return values
  - Reload a module if **syntactically changed** in the IDE



# Automatic Bindings generation

- In the target apps, Lua code interacts heavily with C / ObjC APIs
- May needs to call the System SDK, but also custom APIs defined by the target app
- CodeFlow automatically generates *project* bindings for custom APIs defined in the app's Xcode project.



Select C headers for this bindings library

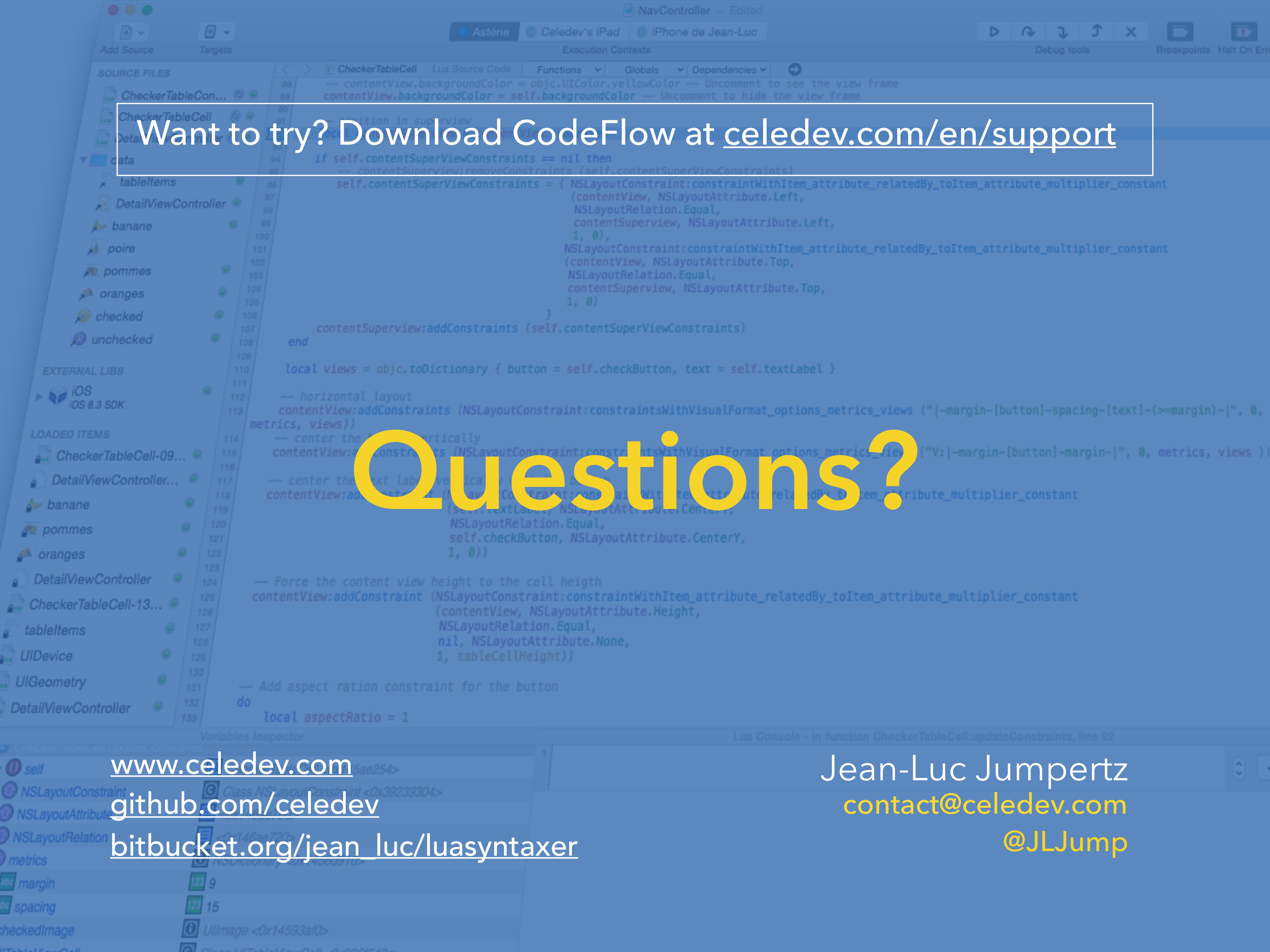
- HelloWorldMac
  - □ AppDelegate.h
  - ☑ ViewController.h

- Integration in the IDE
  - ➡ auto-completion of *project* bindings APIs in the editor
  - ➡ inspection of *project* bindings types in the debugger

# As a conclusion

- The real value of an IDE comes from integration of its components
- A dynamic language like Lua comes with specific challenges for the developer, that a dedicated IDE can help to address.

Want to try? Download CodeFlow at celedev.com/en/support



```
    -- contentView.backgroundColor = objc.UIColor.yellowColor -- Uncomment to see the view frame
    contentView.backgroundColor = self.backgroundColor -- Uncomment to hide the view frame

    if self.contentSuperViewConstraints == nil then
        -- contentSuperview:removeConstraints (self.contentSuperViewConstraints)
        self.contentSuperViewConstraints = { NSLayoutConstraint:constraintWithItem_attribute_relatedBy_toItem_attribute_multiplier_constant
                                             (contentView, NSLayoutAttribute.Left,
                                              NSLayoutRelation.Equal,
                                              contentSuperview, NSLayoutAttribute.Left,
                                              1, 0),
                                            NSLayoutConstraint:constraintWithItem_attribute_relatedBy_toItem_attribute_multiplier_constant
                                            (contentView, NSLayoutAttribute.Top,
                                             NSLayoutRelation.Equal,
                                             contentSuperview, NSLayoutAttribute.Top,
                                             1, 0)
                                           }
        contentSuperview:addConstraints (self.contentSuperViewConstraints)
    end

    local views = objc.toDictionary { button = self.checkButton, text = self.textLabel }

    -- horizontal layout
    contentView:addConstraints (NSLayoutConstraint:constraintsWithVisualFormat_options_metrics_views ("|-margin-[button]-spacing-[text]-(>=margin)-|", 0,
metrics, views))
    -- center the                vertically
    contentView:addConstraints (NSLayoutConstraint:constraintsWithVisualFormat_options_metrics_views ("V:|-margin-[button]-margin-|", 0, metrics, views ))

    -- center the text label vertically
    contentView:addConstraint (NSLayoutConstraint:constraintWithItem_attribute_relatedBy_toItem_attribute_multiplier_constant
                              (self.textLabel, NSLayoutAttribute.CenterY,
                               NSLayoutRelation.Equal,
                               self.checkButton, NSLayoutAttribute.CenterY,
                               1, 0))

    -- Force the content view height to the cell heigth
    contentView:addConstraint (NSLayoutConstraint:constraintWithItem_attribute_relatedBy_toItem_attribute_multiplier_constant
                              (contentView, NSLayoutAttribute.Height,
                               NSLayoutRelation.Equal,
                               nil, NSLayoutAttribute.None,
                               1, tableCellHeight))

    -- Add aspect ration constraint for the button
    do
        local aspectRatio = 1
```

# Questions?

www.celedev.com
github.com/celedev
bitbucket.org/jean_luc/luasyntaxer

Jean-Luc Jumpertz
contact@celedev.com
@JLJump