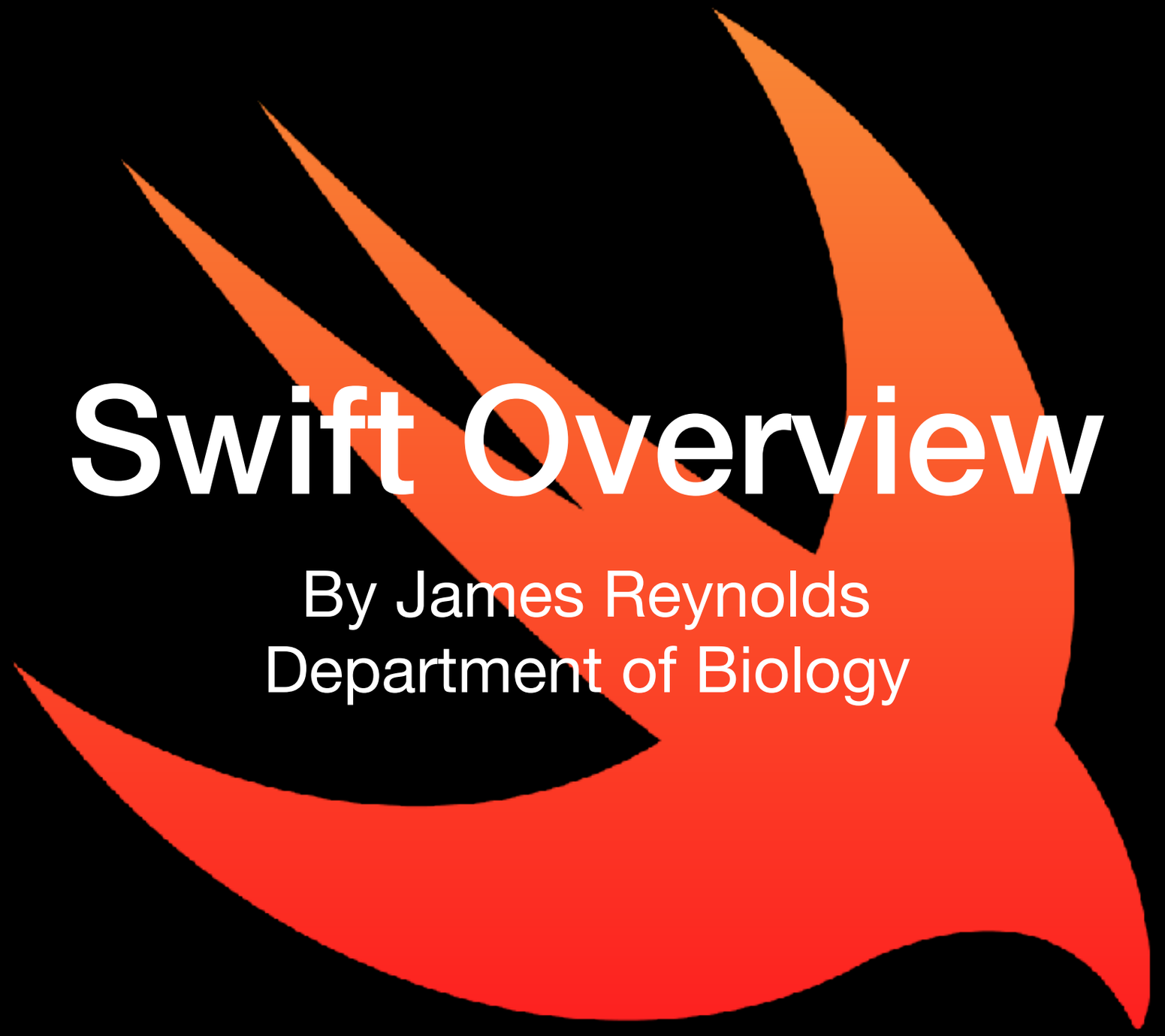


Swift Overview

By James Reynolds
Department of Biology

99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.

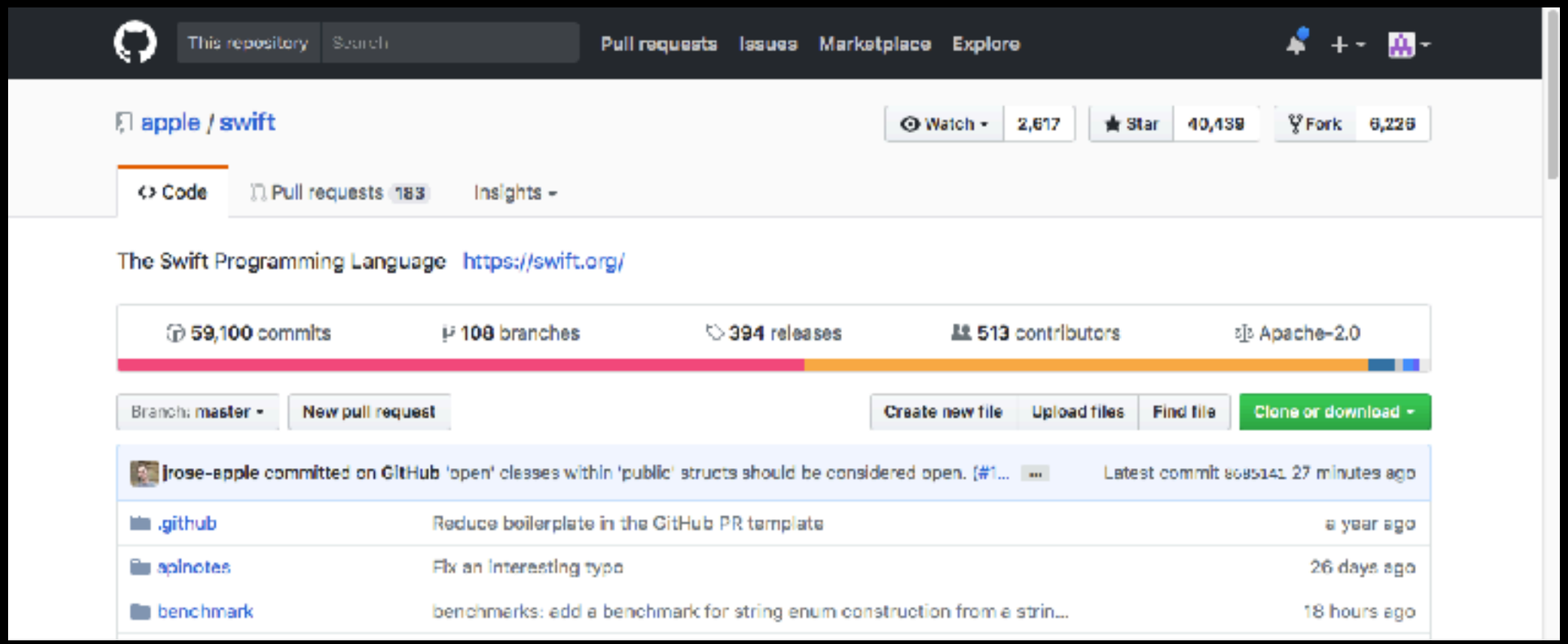
127 little bugs in the code...

What is Swift

- 2014 Announced at WWDC 2014

- “Objective-C without the baggage of C”

What is Swift?



- Open Source on GitHub
- macOS, Linux, FreeBSD

Goals

- Safer than C!
  - No pointers
  - Variables and constants always initialized
  - Array bounds always checked
  - Integer overflow are errors in Swift
  - Memory is managed
  - Objects can’t be nil
  - Xcode debugging
  - Much more

Goals

- Fast (it’s name, haha!)
  - It’s goal is to be as fast as C, C++, and Objective-C
  - Copy on write
- “Protocol-oriented” programming language
  - Use protocols instead of classes and inheritance
- Easy, simple, powerful
  - Lots of shorthands, modernisms, syntactic sugar

Chris Lattner

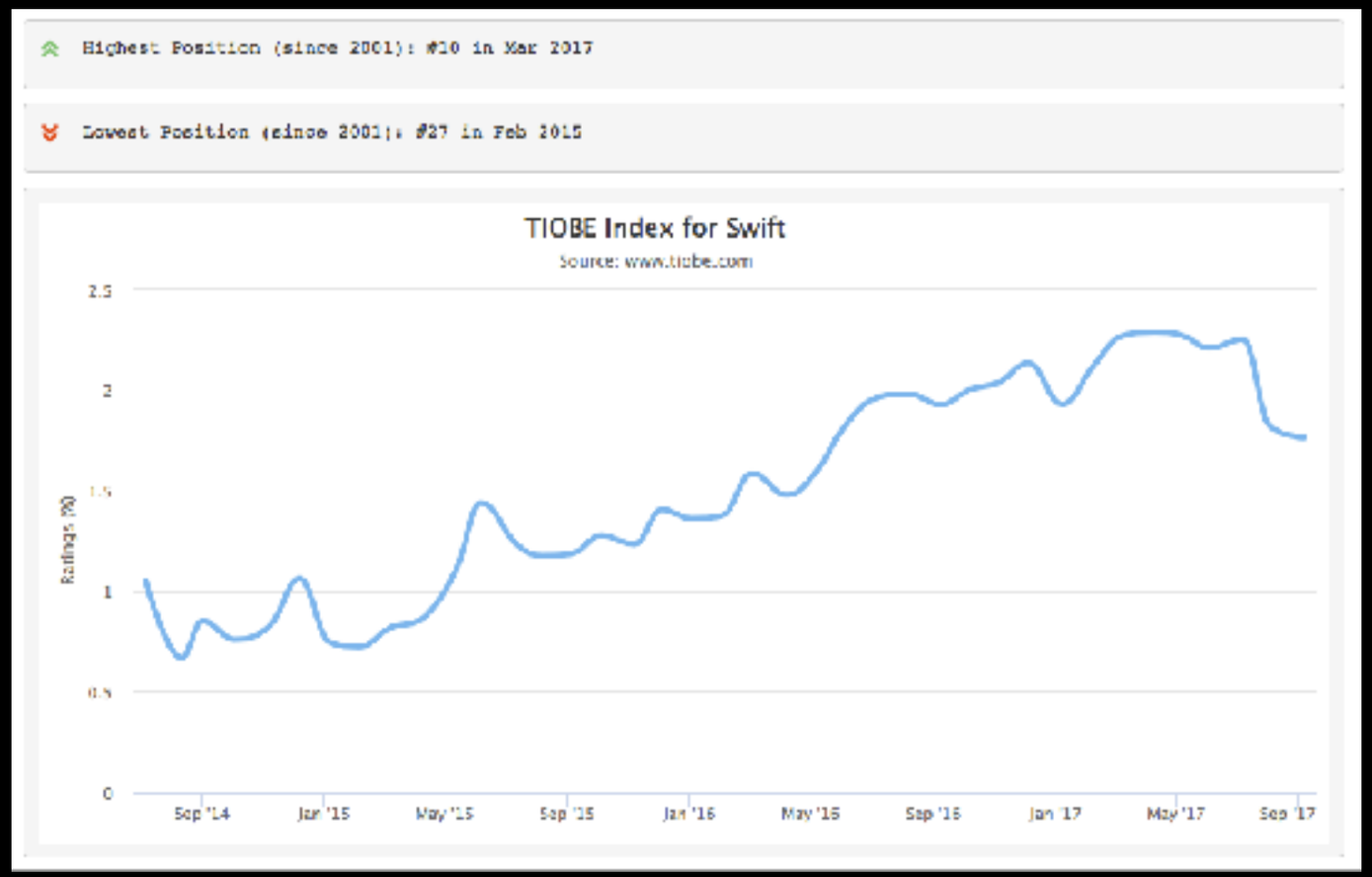
- Chris Lattner created LLVM at University of Illinois at Urbana-Champaign
- Lattner brought LLVM to Apple in 2005
- Chris Lattner started Swift in 2010
- In January 2017, Lattner left Apple

Popularity

- Swift replacing Objective-C for macOS and iOS dev
- Won “Most Loved Programming Language” on Stack Overflow in 2015, and got 2nd place in 2016

Popularity

- In March 2017 Swift made the top 10 TIOBE index
- Some say it has the same popularity as JavaScript, PHP, and C#
- Did I mention it’s only 7 (3) years old?



Popularity

- iOS dev
- Objective-C
- Swift

```
- (void)viewDidLoad
{
    EAGLContext *aContext = [[EAGLContext alloc] initWithAPI:

    if (!aContext)
        NSLog(@"Failed to create ES context");
    else if (![EAGLContext setCurrentContext:aContext])
        NSLog(@"Failed to set ES context current");

    self.context = aContext;
    [aContext release];

    [(EAGLView *)self.glView setContext:context];
    [(EAGLView *)self.glView setFramebuffer];
}
```

Versioning

- Swift is rapidly changing and versions are incompatible
- 2014: Swift 1 (Xcode 6, OS X 10.9.3-10.10, iOS 7-8)
- 2015: Swift 2 (Xcode 7, 10.10.4-10.11, iOS 9)
- 2016: Swift 3 (Xcode 8, 10.11.5-10.12, iOS 10)
- 2017: Swift 4 and 3.2 (Xcode 9, 10.12.6+, iOS 11)

Size of an App

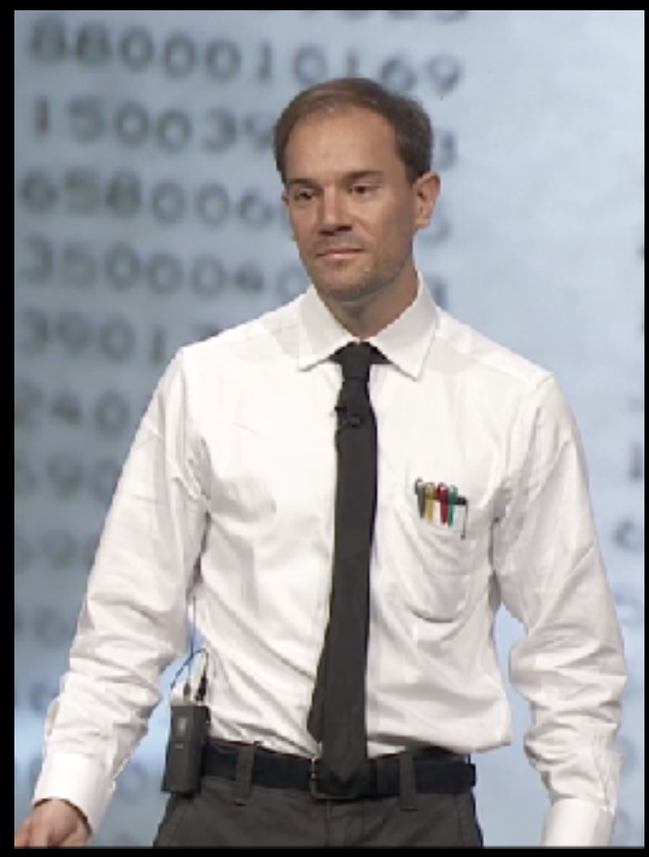
- Swift libraries are included in the app instead of the OS
- Size of app: 10 MB
  - Frameworks: 8.9 MB (Swift 4 reduces this)
  - Icon: .8 MB
  - Rest of app: .3 MB

Features

- Influenced by C#, CLU, D, Haskell, Objective-C, Python, Ruby, Rust, more
- Multi-paradigm: protocol-oriented, object-oriented, functional, imperative, block structured
- Typing: Static, strong, inferred
- Closures, tuples, maps and filters
- Generics, reflective, event-driven concurrent, and failsafe
- Not procedural or standardized
- Package manager

Playgrounds

- Swift Playgrounds on iPad
- Xcode Playground projects
    - Live coding à la Bret Victor
- Online IBM REPL Sandbox for Linux
    - https://swift.sandbox.bluemix.net/#/repl



Balloons — Balloons.playground — Edited

Balloons.playground › setupHero(_:_:)

```
func doDidMoveToView(scene : SKScene,
                     delegate : SKPhysicsContactDelegate) {


    // ================== Blimp Control ====================

    yOffsetForTime = { i in
        return 80 * sin(i / 10.0)
    }

    // ============= Scene Configuration ====================

    // Set up balloon lighting and per-pixel collisions.
    balloonConfigurator = { b in
        b.physicsBody.categoryBitMask = CONTACT_CATEGORY
        b.physicsBody.fieldBitMask = WIND_FIELD_CATEGORY
        b.lightingBitMask = BALLOON_LIGHTING_CATEGORY
    }

    // Load images for balloon explosion.
    balloonPop = (1...4).map {
        SKTexture(imageNamed: "explode_0\($0)")
    }

    // Install turbulant field forces.
    var turbulence = SKFieldNode.noiseFieldWithSmoothness(0.7,
                                 animationSpeed:0.8)
    turbulence.categoryBitMask = WIND_FIELD_CATEGORY
    turbulence.strength = 0.21
    scene.addChild(turbulence)

    cannonStrength = 210.0

    // ================ Scene Initialization =================

    // Do the rest of the setup and start the scene.
    setupHero(scene, delegate)
    setupFan(scene, delegate)
    setupCannons(scene, delegate)
}


func handleContact(bodyA : SKSpriteNode,
                   bodyB : SKSpriteNode) {

    if (bodyA == hero) {
        bodyB.normalTexture = nil
        bodyB.runAction(removeBalloonAction)
    } else if (bodyB == hero) {
        bodyA.normalTexture = nil
        bodyA.runAction(removeBalloonAction)
    }
}
```

(Function)
(1058 times)

(Function)

(55 times)

[SKTexture, SKTexture, SKTe...
(4 times)

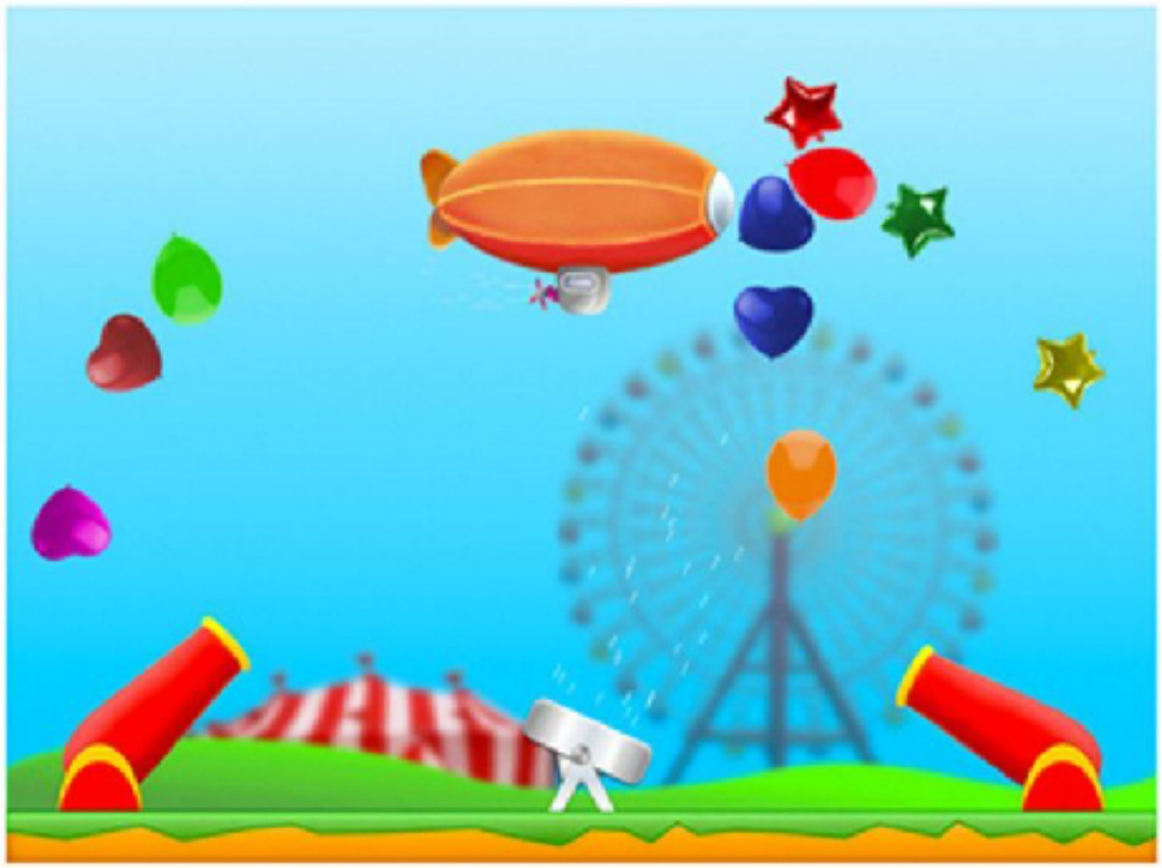
SKNoiseFieldNode

SKNoiseFieldNode
SKNoiseFieldNode
(GameScene ((Function)) ((F...

210.0

Timeline › Balloons.playground (Timeline)

Balloons



let y = 80 * sin(x)

#!/usr/bin/swift

- Do not confuse with the Swift Parallel Scripting Language
- Pros
- C API in a “script”
- Swift is Apple’s future
- Xcode IDE

#!/usr/bin/swift

- Cons
  - Xcode must be installed to run it
    - Without it `/usr/bin/swift` produces

```
xcode-select: note: no developer tools were found at '/Applications/
Xcode.app', requesting install. Choose an option in the dialog to
download the command line developer tools.
```

  - Must agree the Xcode license
  - Not exactly portable or backwards compatible
  - Language is still only 7 (3) years old

Syntax

Comments and Vars

```
/* Swift multiline

comment /* can */ nest */

var age = 25 // inferred Int

var salary : Float // explicit uninitialized Float

let pi = 3.1415 // Constant

var msg = "An NSString in Swift"

var s = msg.count // s is 17 (4.0 syntax)
```

Arrays

```
var nums = [1,2,3,5,7,11] // initialize

var strings = ["hello", "how are you", "goodbye"]

var num = nums[0] // access value: 1

nums[2] = 10 // change value

nums.append(7) // add value

for n in nums { // iterate

    print(n)

}
```

Dictionaries

```
var cust = ["name": "Brad", "state": "CA"] // initialize

var the_name = cust["name"] // access value

cust["name"] = "Bradley" // change value

cust["hobby"] = "Swimming" // add key/value

cust.removeValue(forKey: "hobby") // remove key/value

for (key, value) in cust { // iterate

    print("\(key) = \(value)")

}
```

Switch

```
switch (i) {
    case 1...3:
        print(“1...3") // supports ranges
    case 4:
        fallthrough // must explicitly fall through
    default:
        print(“default")
}
```

Functions

```
func getUser(id: Int) -> (String, String) {
    var username = "username"
    var email = "email"
    return (username, email)
}
var (u, e) = getUser(id:1)
print("\(u):\(e)")
```

Classes

```
class ViewController: NSViewController {
  @IBOutlet weak var timerField: NSTextField!

  override func viewDidLoad() {
    super.viewDidLoad()
    let window: NSWindow = NSApp.windows[0]
    window.center()
    window.makeKeyAndOrderFront(self)
  }

}
```

Example

Questions?