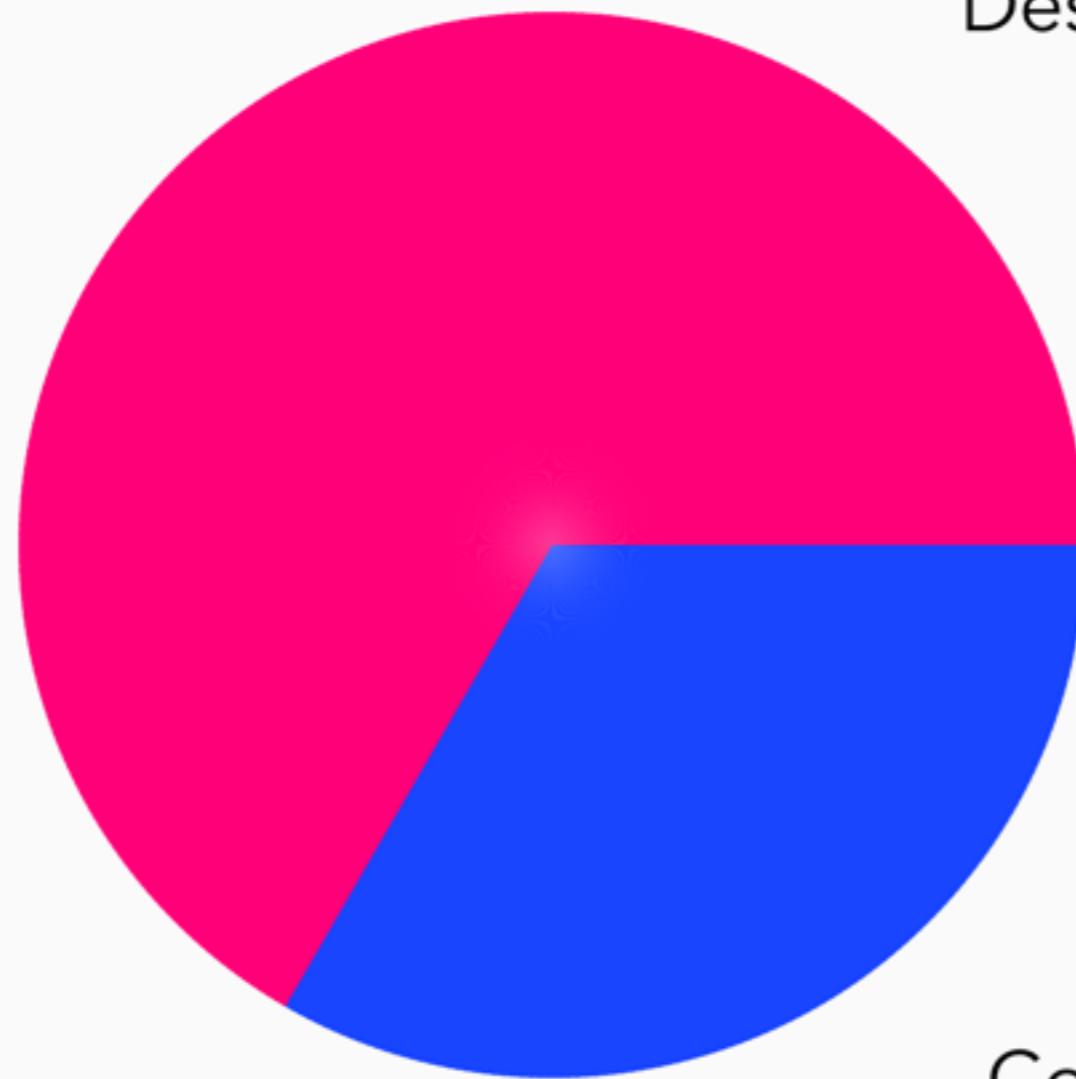




Code, creatively.

What

Design



Code

# How

```
let frame = CGRect(x: 0, y: 0, width: 100, height: 100)
let shapeView = UIView(frame: frame)
let shapeLayer = CAShapeLayer()
let path = CGPathCreateWithEllipseInRect(shapeView.bounds,
nil)
shapeLayer.fillColor = UIColor.blueColor().CGColor
shapeLayer.path = path
shapeView.layer.addSublayer(shapeLayer)
view.addSubview(shapeView)
```

# How

```
let circle = Circle(center: location, radius: 1)  
self.canvas.add(circle)
```

# How

```
UIView.animateWithDuration(0.5) {
    circle.center = CGPointMake(x: circle.center.x + 100, y:
circle.center.y)
}

CATransaction.begin()
CATransaction.setValue(NSNumber(float: 0.5), forKey:
kCATransactionAnimationDuration)
if let shapeLayer = circle.layer as? CAShapeLayer {
    shapeLayer.lineWidth = 20.0
}
CATransaction.commit()
```

# How

```
ViewAnimation(duration: 0.5) {  
    circle.center += Vector(x: 100, y: 0)  
    circle.lineWidth = 20.0  
}
```

# How

```
UIView.animateWithDuration(0.25, animations: {
    circle.center = CGPointMake(x: circle.center.x + 100, y:
circle.center.y)
    }, completion: { completed in
        circle.removeFromSuperview()
})
```

```
UIView.animateWithDuration(0.5) {
    circle.center = CGPointMake(x: circle.center.x + 100, y:
circle.center.y)
}
```

```
CATransaction.begin()
CATransaction.setValue(NSNumber(float: 0.5), forKey:
kCATransactionAnimationDuration)
if let shapeLayer = circle.layer as? CAShapeLayer {
    shapeLayer.lineWidth = 20.0
}
CATransaction.commit()
```

# How

```
let a = ViewAnimation(duration: 0.25) {  
    circle.center += Vector(x: 100, y: 0)  
    circle.lineWidth = 20.0  
}  
  
a.addCompletionObserver() {  
    circle.removeFromSuperview()  
}
```

# How

```
func viewDidLoad() {
    super.viewDidLoad()
    let pan = UIPanGestureRecognizer(target: self, action:
#selector(handlePan))
    view.addGestureRecognizer(pan)
}

func handlePan(gesture: UIPanGestureRecognizer) {
    switch gesture.state {
    case .Began:
        var locations = [CGPoint]()
        for i in 0..
```

# How

```
func setup() {  
    canvas.addPanPressGestureRecognizer { locations, center,  
state in  
        switch state {  
        case .Began:  
            print(locations)  
        case .Changed:  
            print(center)  
        default:  
            break  
        }  
    }  
}
```

# How

```
canvas.addPanGestureRecognizer { (locations, center,
translation, velocity, state) in
    if state == .Began {
        for location in locations {
            let c = Circle(center: location, radius: 25)
            self.canvas.add(c)

            let a = ViewAnimation(duration: 0.5) {
                c.center += Vector(x: velocity.x / 10.0, y:
velocity.y / 10.0)
                c.opacity = 0.0
                c.lineWidth = 50.0
            }

            a.addCompletionObserver() {
                c.removeFromSuperview()
            }
            a.animate()
        }
    }
}
```

# How

```
func viewDidLoad() {
    guard let url = NSBundle.mainBundle().URLForResource("halo.mp4",
withExtension: nil) else {
        fatalError("File not found")
}

let asset = AVAsset(URL: url)
let player = AVQueuePlayer(playerItem: AVPlayerItem(asset: asset))
player.actionAtItemEnd = .Pause

let movieLayer = AVPlayerLayer(player: player)
movieLayer.videoGravity = AVLayerVideoGravityResize

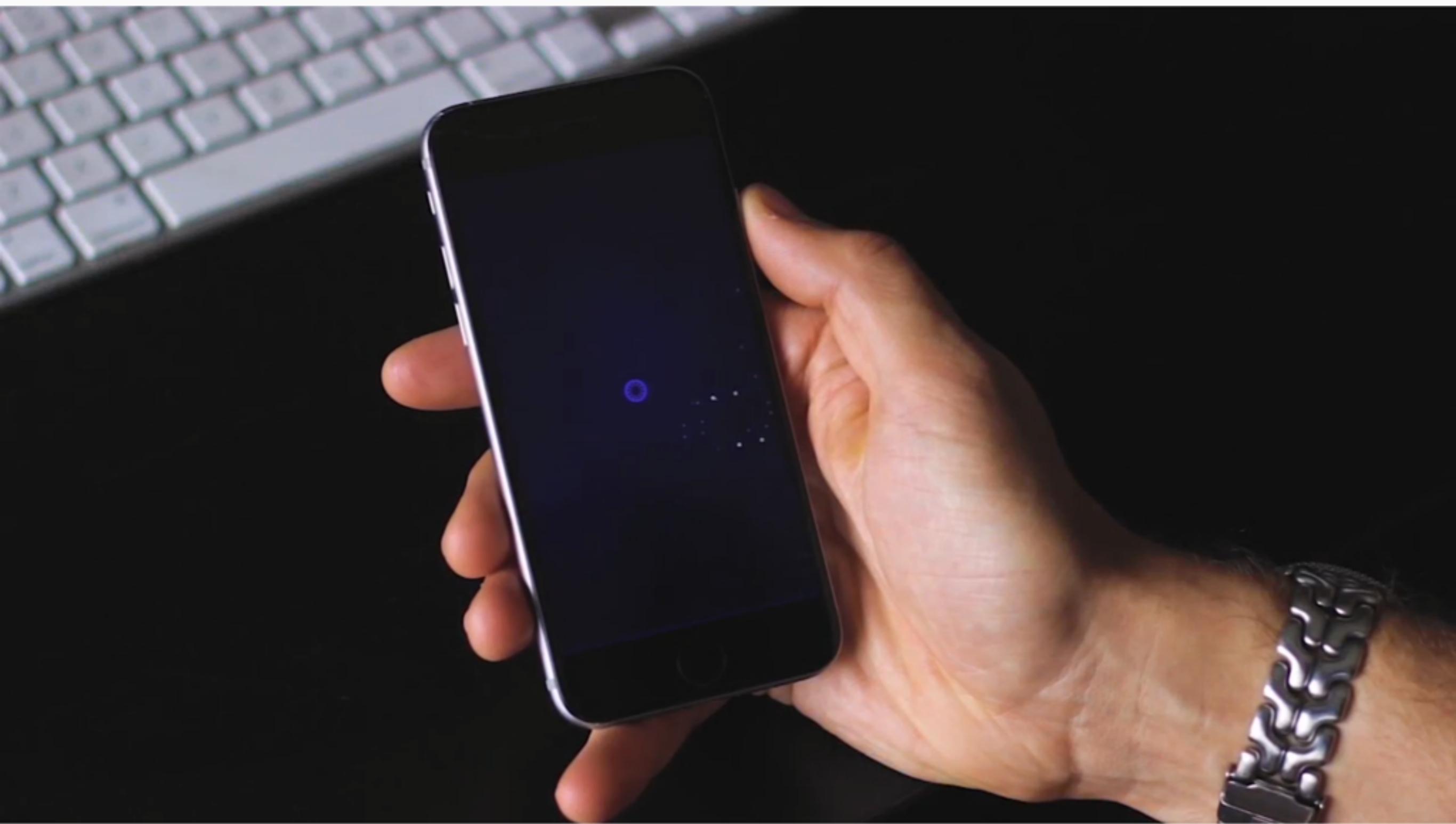
let tracks = asset.tracksWithMediaType(AMediaTypeVideo)

let movieTrack = tracks[0]
let size = movieTrack.naturalSize

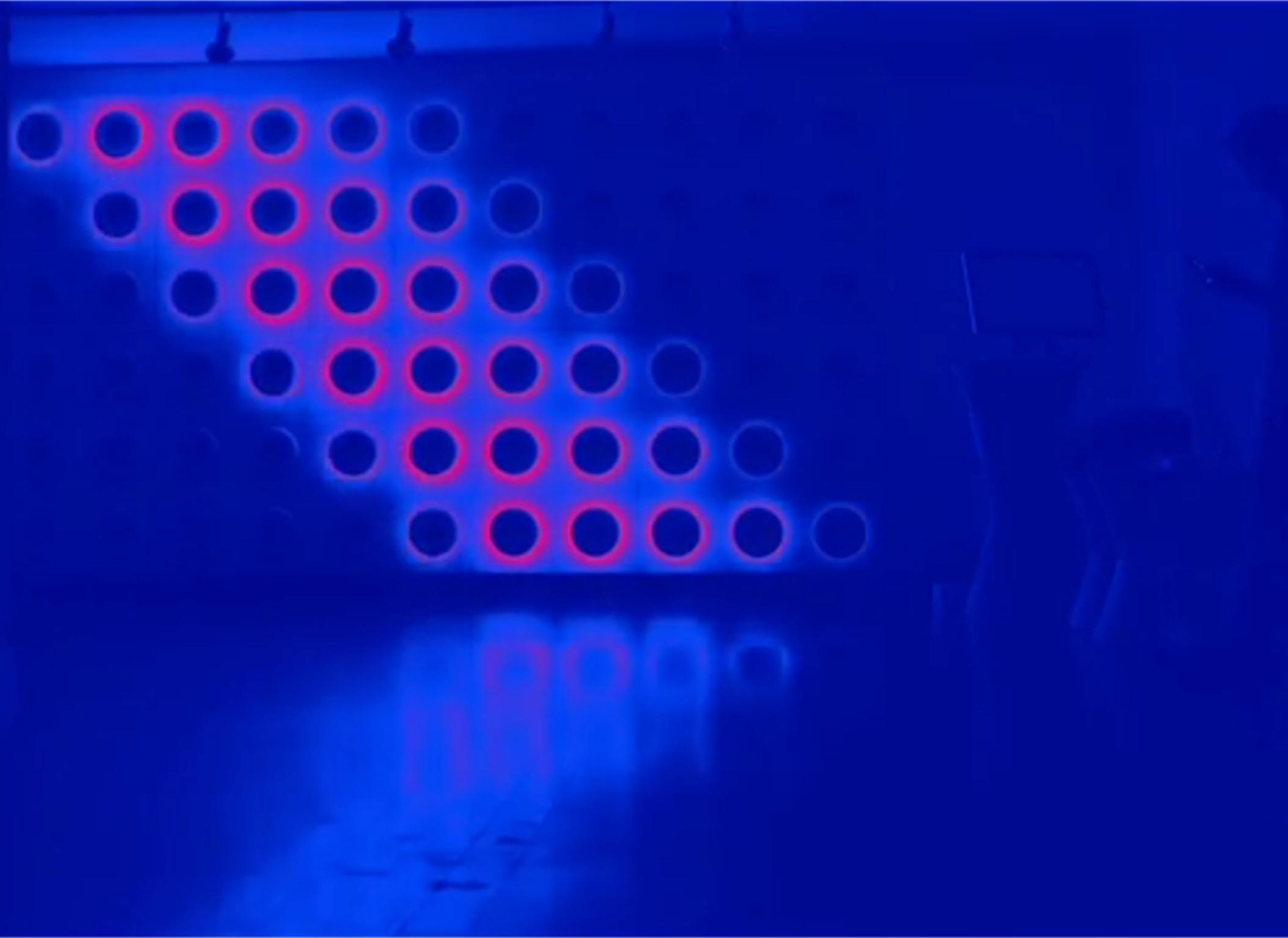
movieLayer.frame = CGRect(x: 0,y: 0,width: size.width,height:
size.height)
self.view.layer.addSublayer(movieLayer)
player.play()
}
```

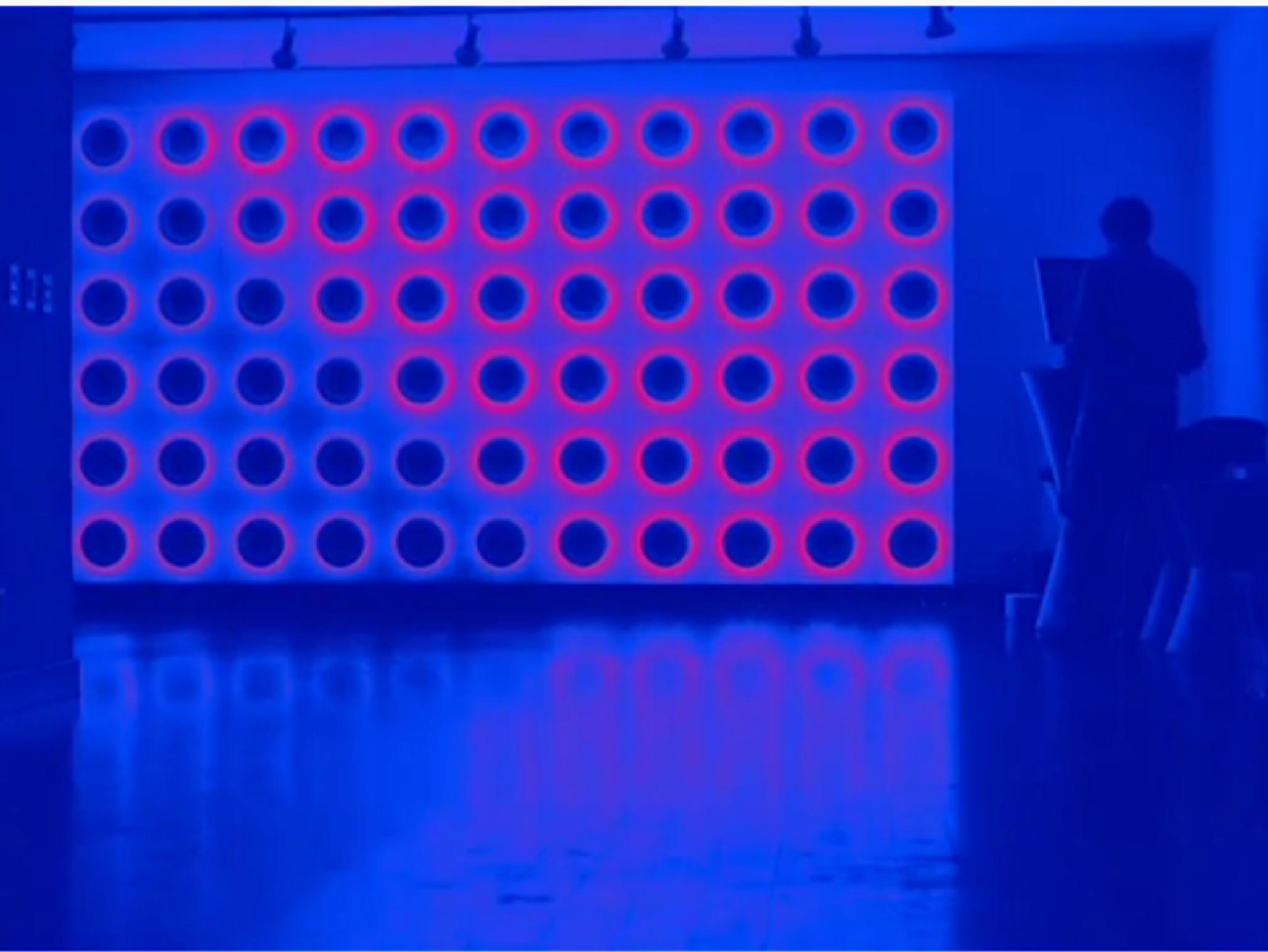
# How

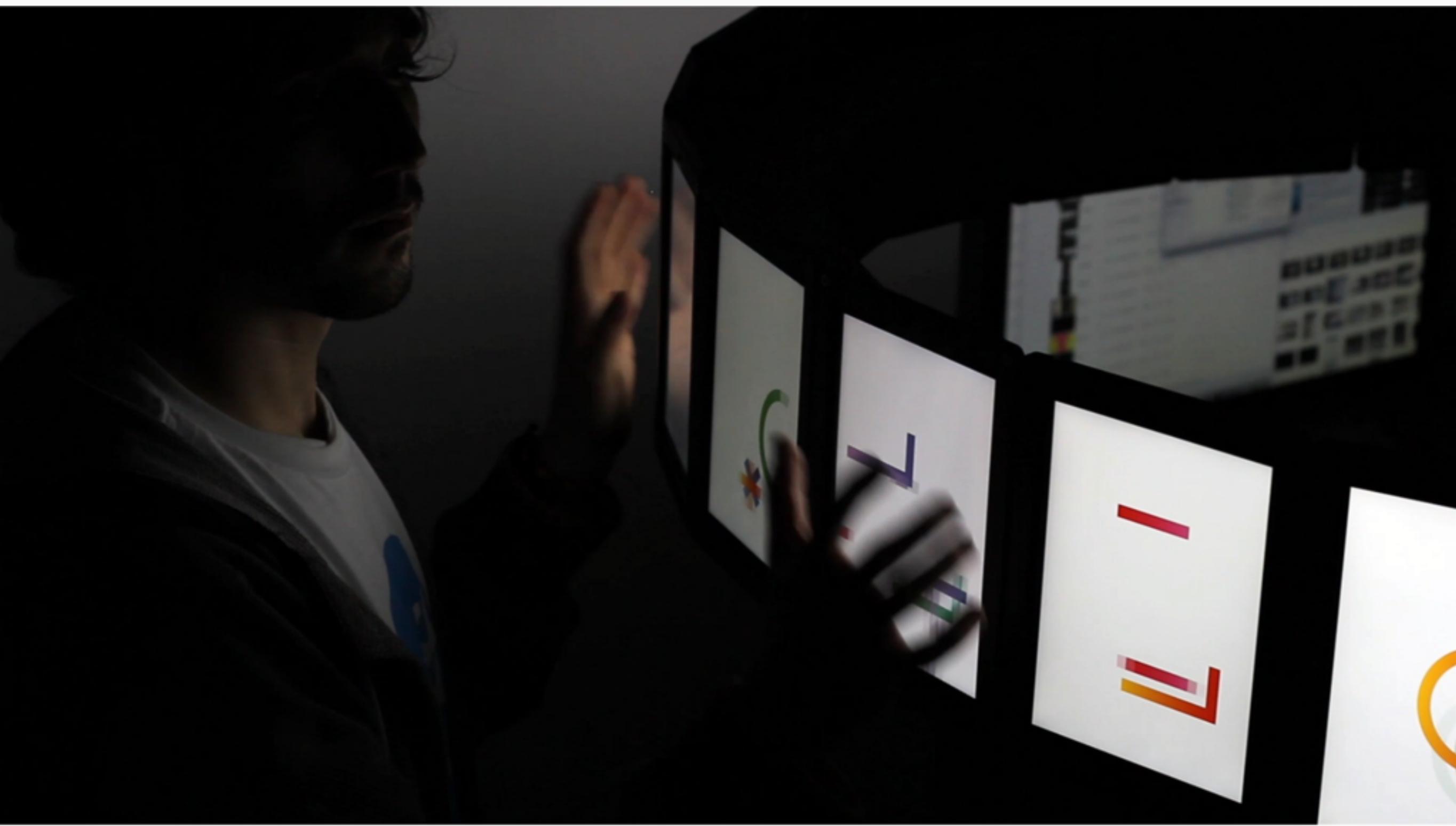
```
func setup() {  
    let movie = Movie("halo.mp4")  
    canvas.add(movie)  
    movie?.play()  
}
```







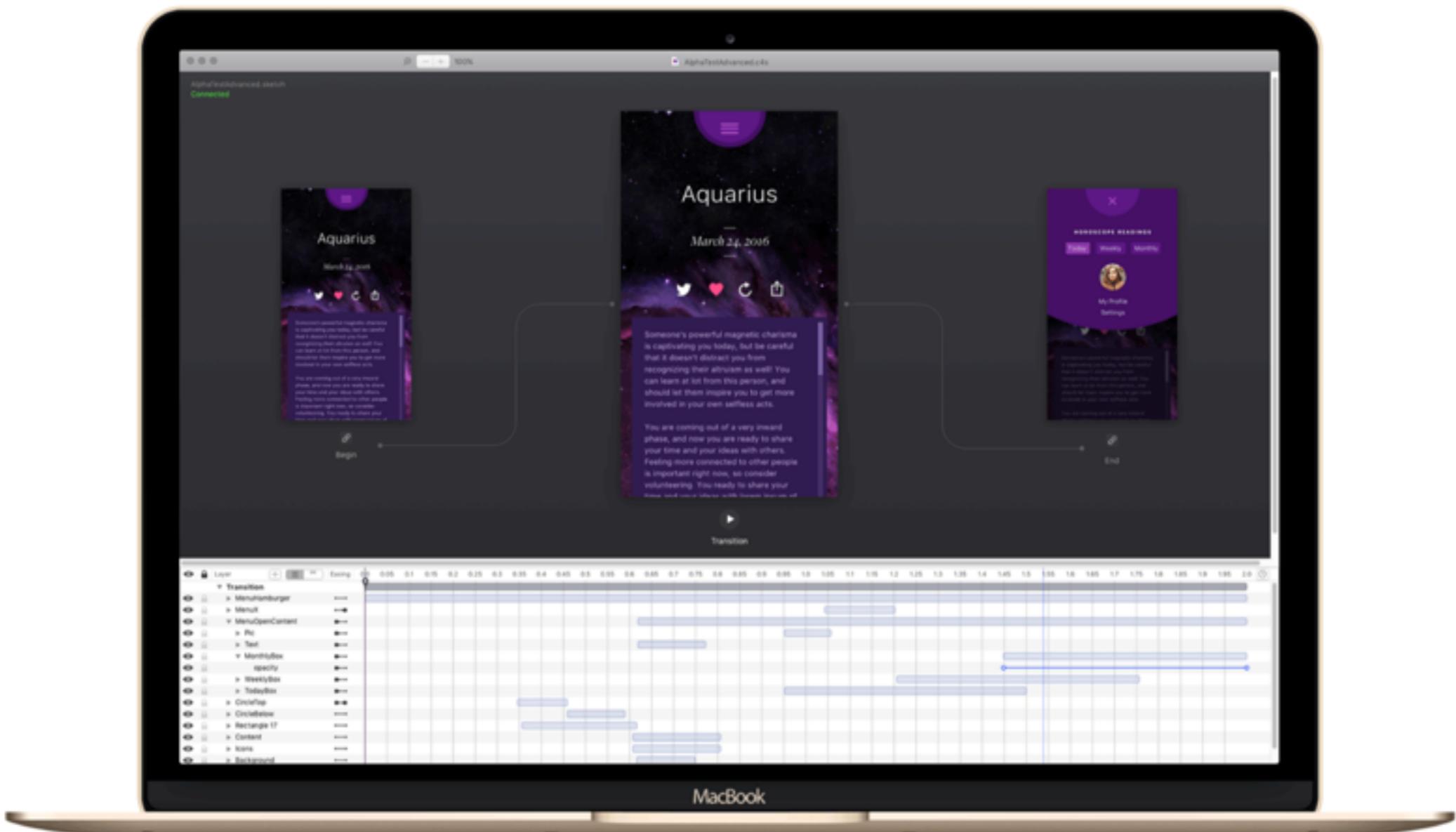




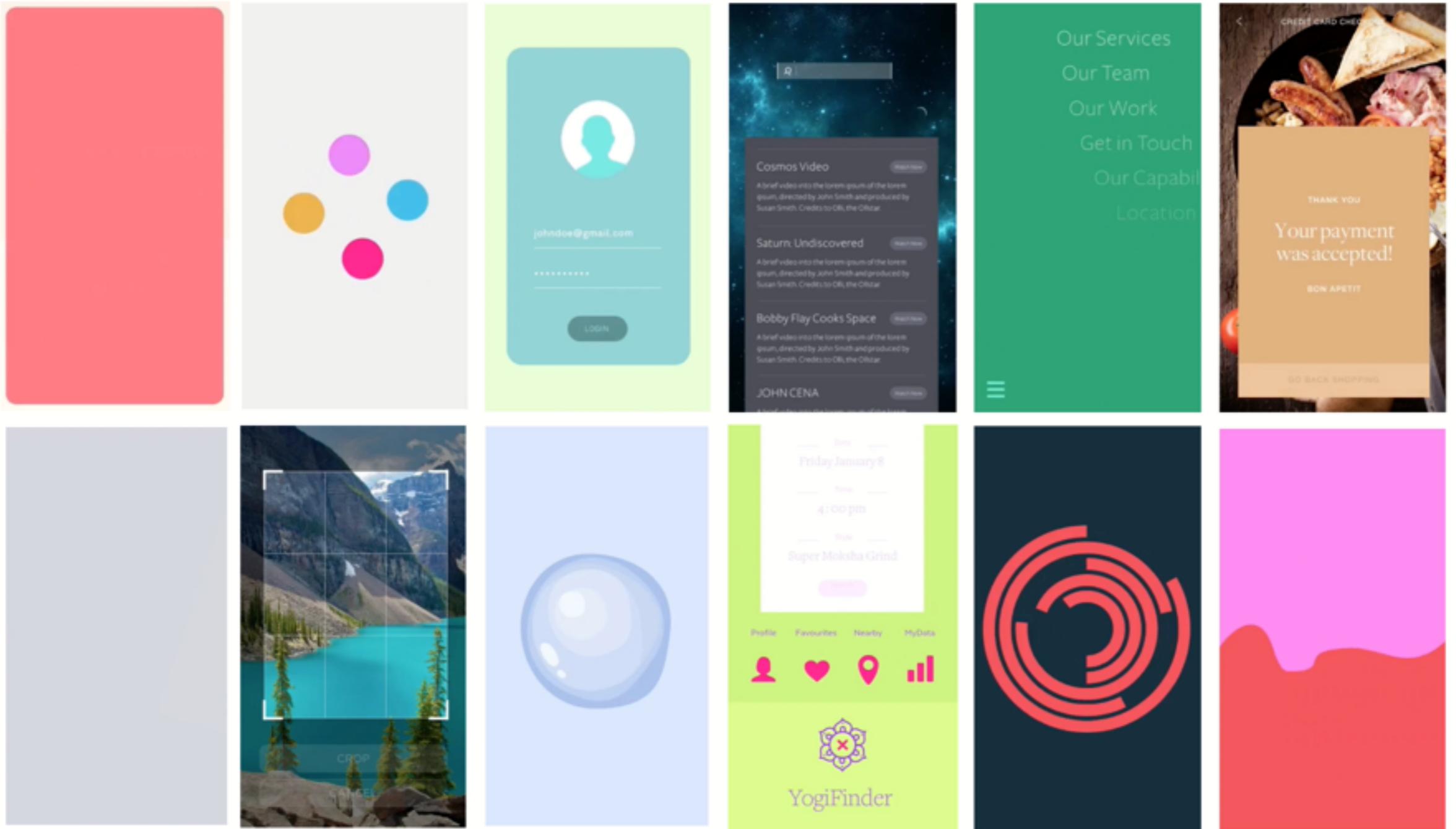


[www.c4ios.com](http://www.c4ios.com)









# More

- [www.c4ios.com](http://www.c4ios.com)
- [www.c4studio.co](http://www.c4studio.co)
- <https://github.com/C4Framework/C4iOS>
- [www.logicandform.com](http://www.logicandform.com)