



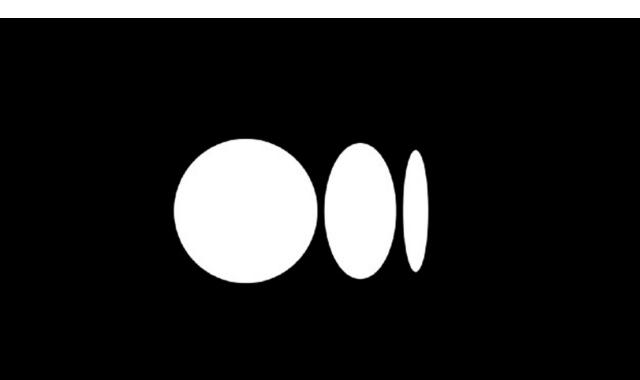
# Fine I'll draw it myself

## How to create custom Composables

Alexander Frank @ OneCode GmbH

# About me

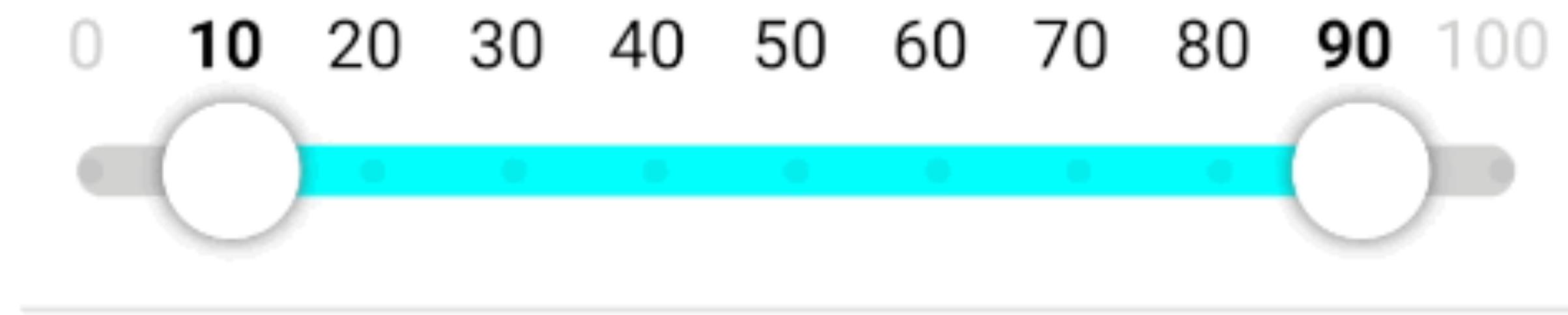
- Software Engineer since 2006
- Android development since 2012
- Working for OneCode GmbH in Germany
- Workation for two months in Portugal



OneCode

# Custom Composable

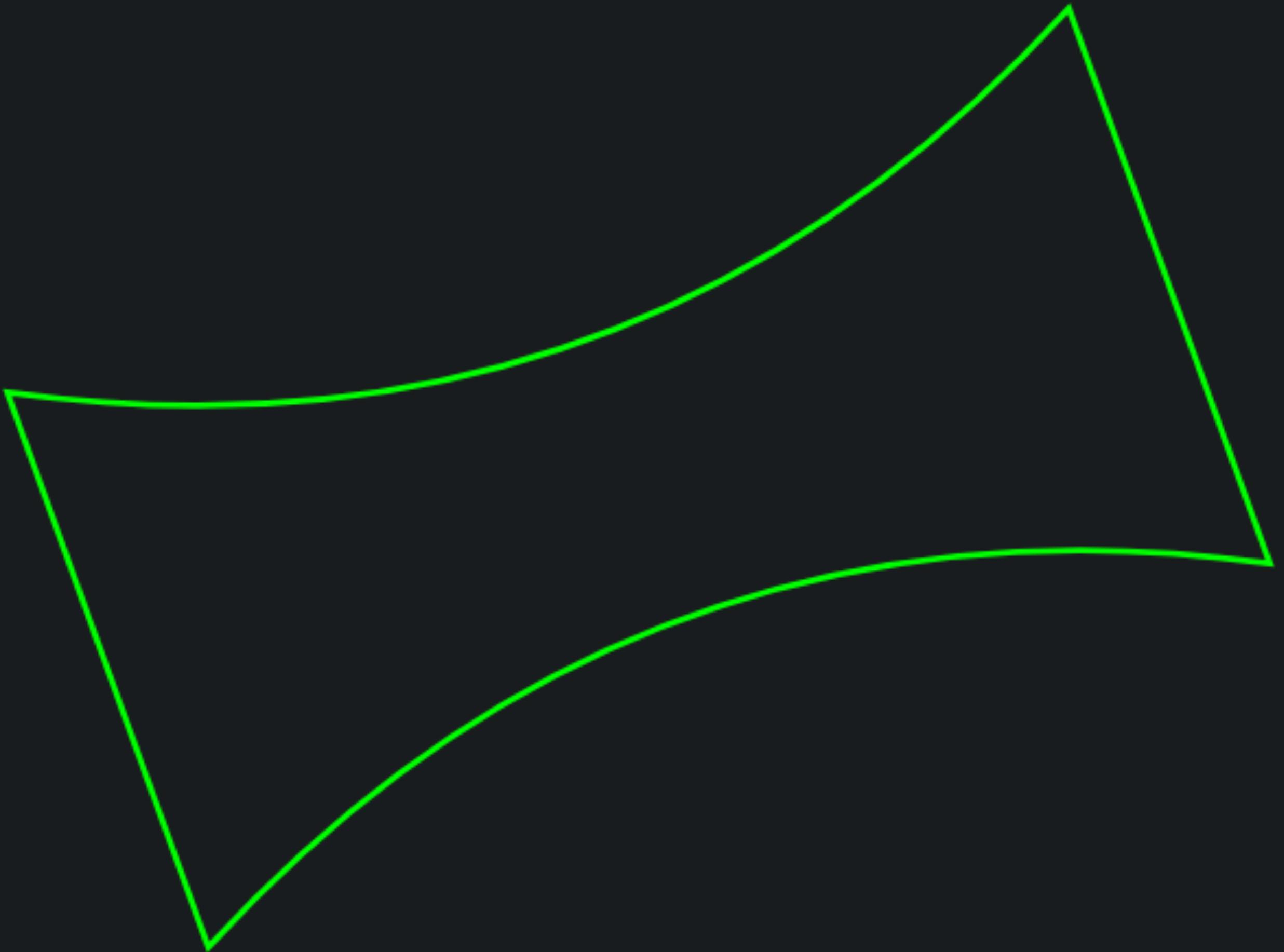
- Complex UI elements
- Not easily composable
- Exact relativ positioning
- Part of a bigger layout



The selected range is 10..90

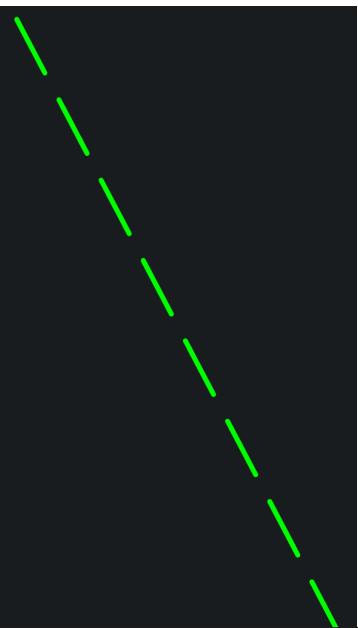
# Canvas

- Composable with DrawScope
- Provides many draw functions
- Measurements in Pixels
- Scale/Rotation/Translation

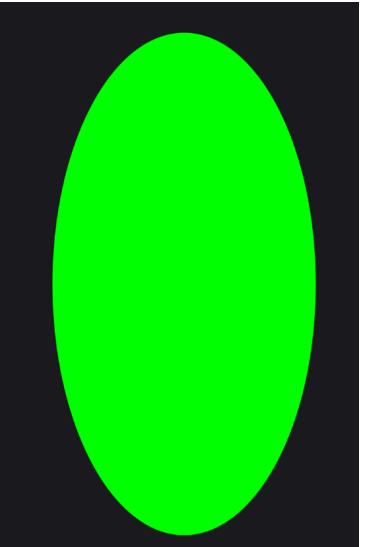


# Demo

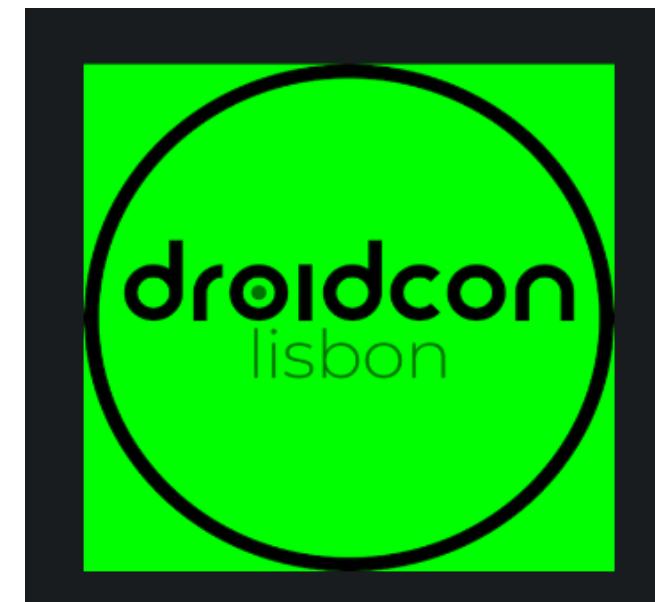
# Additional drawing options



```
drawLine(  
    color = Color.Green,  
    start = Offset(quarterCanvasWidth, quarterCanvasHeight),  
    end = Offset(x: halfCanvasWidth + quarterCanvasWidth, y: halfCanvasHeight + quarterCanvasHeight),  
    strokeWidth = 3.dp.toPx(),  
    cap = StrokeCap.Round,  
    pathEffect = PathEffect.dashPathEffect(floatArrayOf(100f, 50f))  
)
```

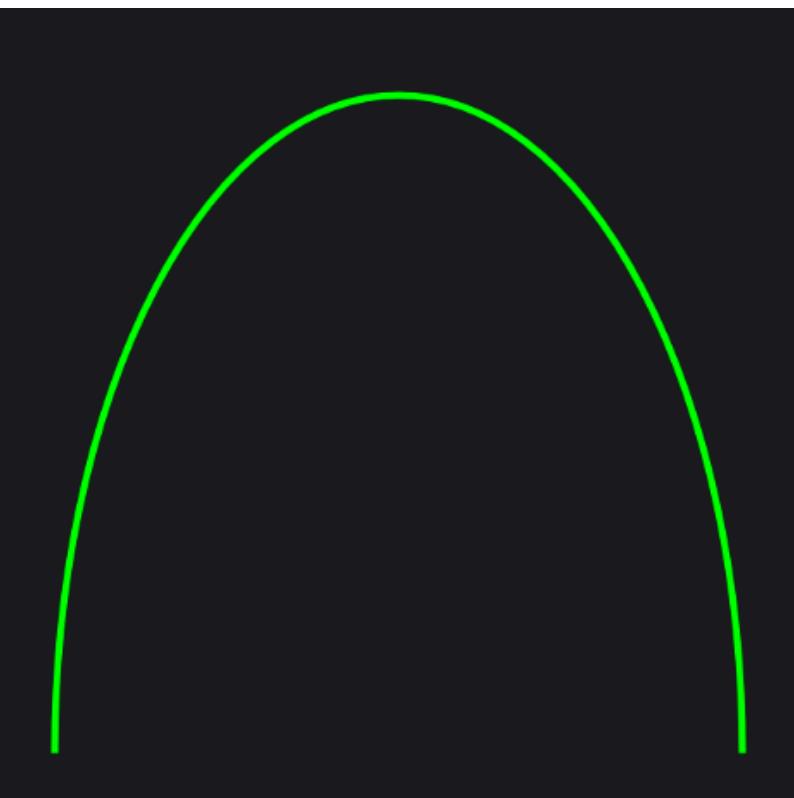


```
drawOval(  
    color = Color.Green,  
    topLeft = Offset(quarterCanvasWidth, quarterCanvasHeight),  
    size = Size(halfCanvasWidth, halfCanvasHeight)  
)
```

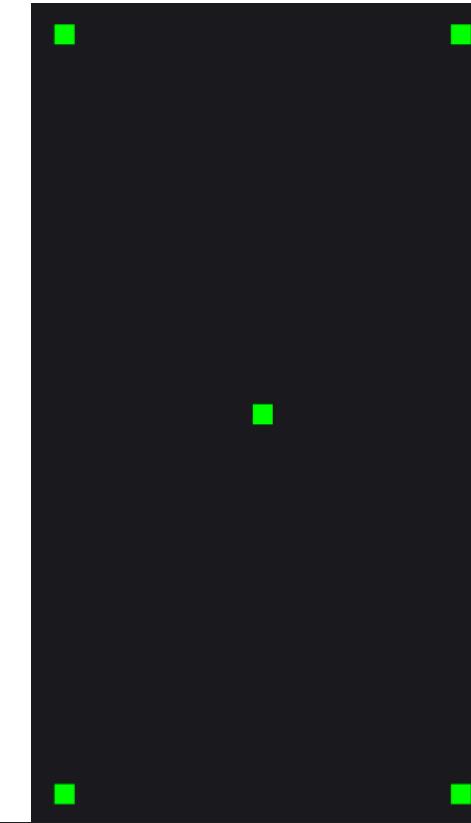


```
drawImage(  
    image = logo,  
    dstSize = IntSize(halfCanvasWidth.toInt(), quarterCanvasHeight.toInt()),  
    dstOffset = IntOffset(quarterCanvasWidth.toInt(), quarterCanvasHeight.toInt()),  
    colorFilter = ColorFilter.lighting(Color.Green, Color.Black)  
)
```

# Additional drawing options



```
drawArc(  
    color = Color.Green,  
    topLeft = Offset(quarterCanvasWidth, quarterCanvasHeight),  
    size = Size(halfCanvasWidth, halfCanvasHeight),  
    startAngle = 180f,  
    sweepAngle = 180f,  
    useCenter = false,  
    style = Stroke(2.dp.toPx())  
)
```



```
drawPoints(  
    color = Color.Green,  
    points = listOf(  
        Offset(quarterCanvasWidth, quarterCanvasHeight),  
        Offset(halfCanvasWidth, halfCanvasHeight),  
        Offset(x: canvasWidth - quarterCanvasWidth, quarterCanvasHeight),  
        Offset(x: canvasWidth - quarterCanvasWidth, y: canvasHeight - quarterCanvasHeight),  
        Offset(quarterCanvasWidth, y: canvasHeight - quarterCanvasHeight),  
    ),  
    pointMode = PointMode.Points,  
    strokeWidth = 10.dp.toPx()  
)
```

3:28



# Touchable

# **Modifier.pointerInput**

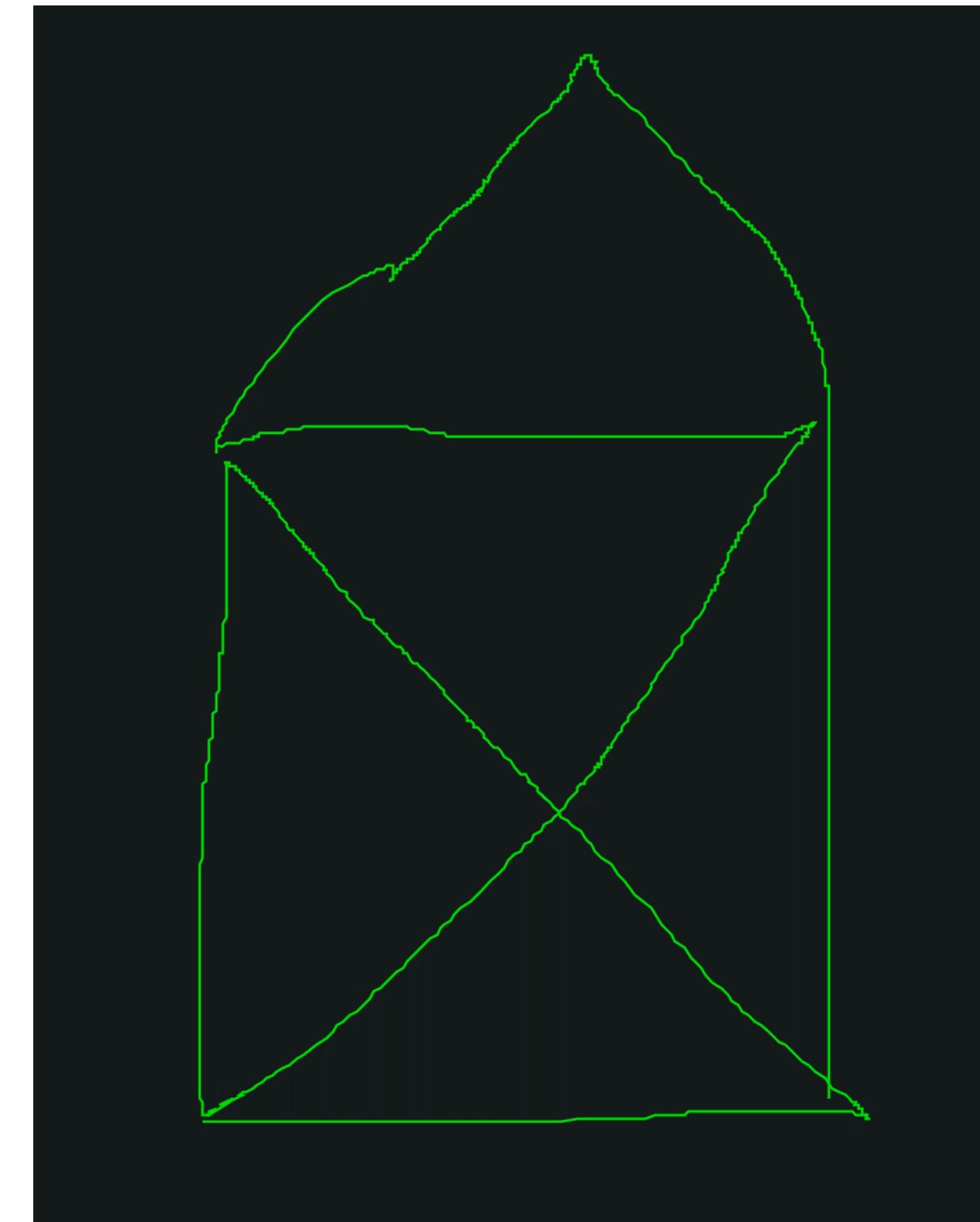
## **Predefined detectors**

- detectTapGestures
- detectDragGestures
- detectHorizontalDragGestures
- detectVerticalDragGestures
- detectDragGesturesAfterLongPress
- detectTransformGestures

# Modifier.pointerInput

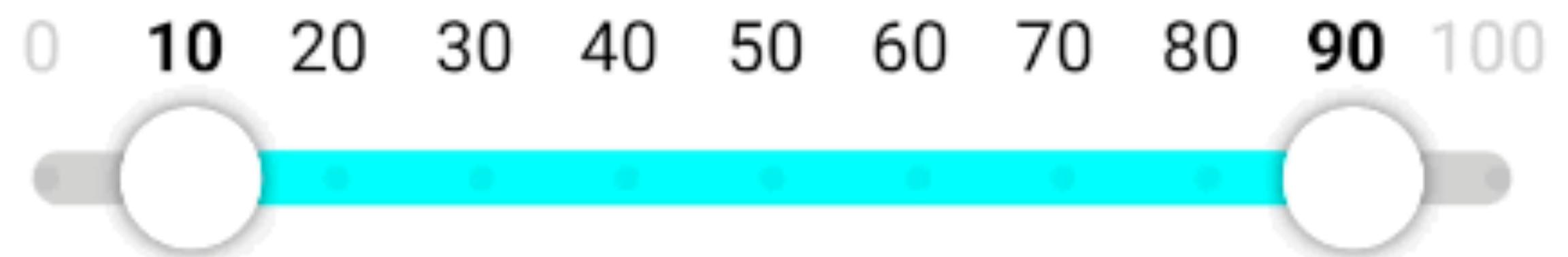
## Custom handling

- awaitEachGesture

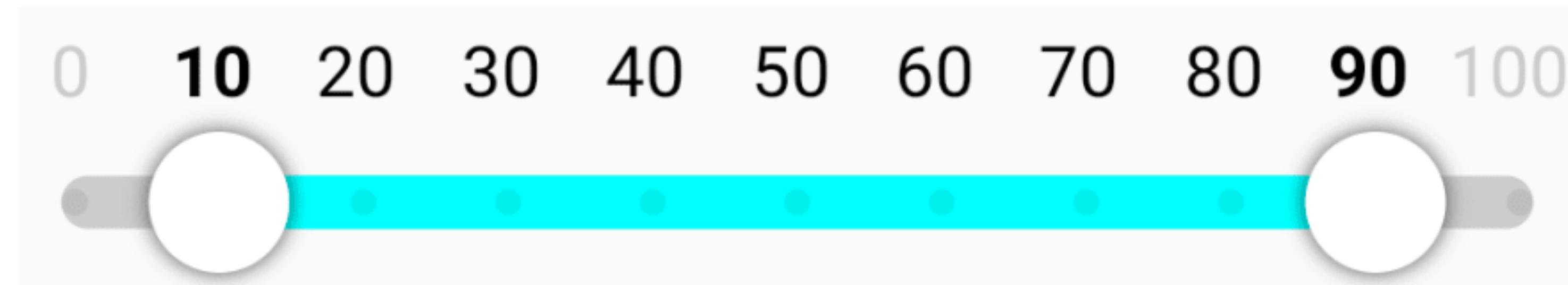


# Demo

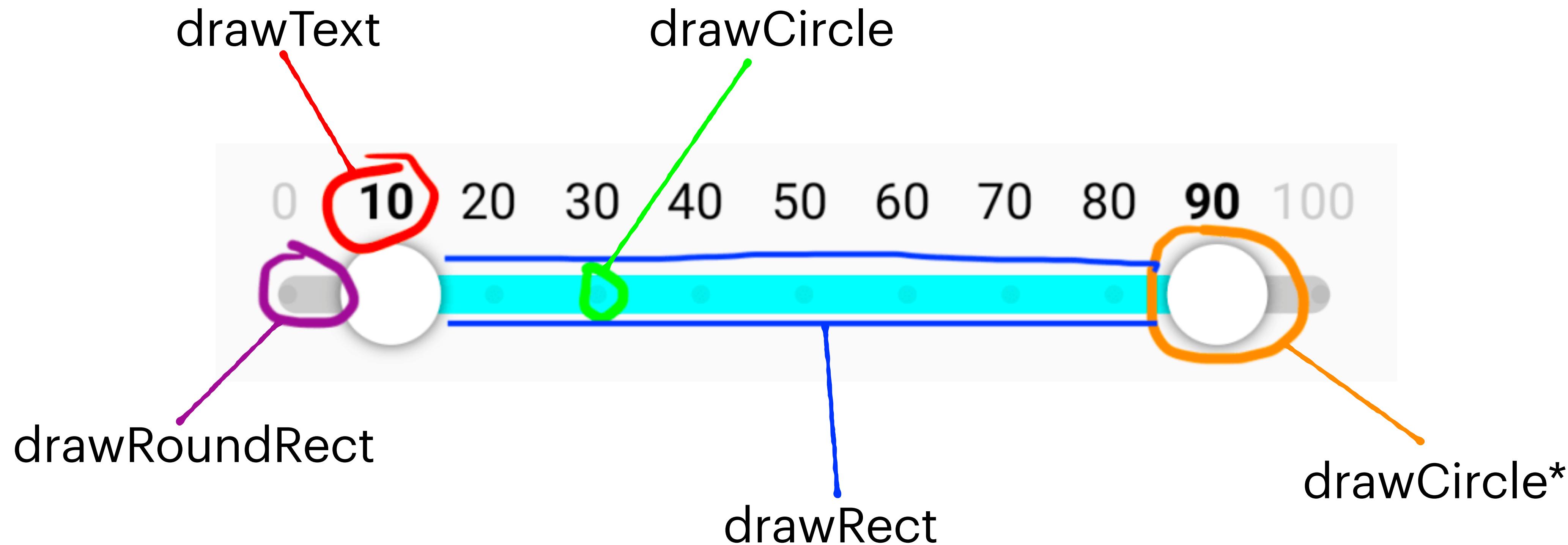
# Labeled Range Slider



# Divide and Conquer



# Divide and Conquer



# Demo

# Questions?

## Canvas

- draw...
- toPx
- transform/rotate/scale
- drawIntoCanvas

## Modifier.pointerInput

- detect...
- awaitEachGesture

