Understanding Constraints, Relationships, and Priorities

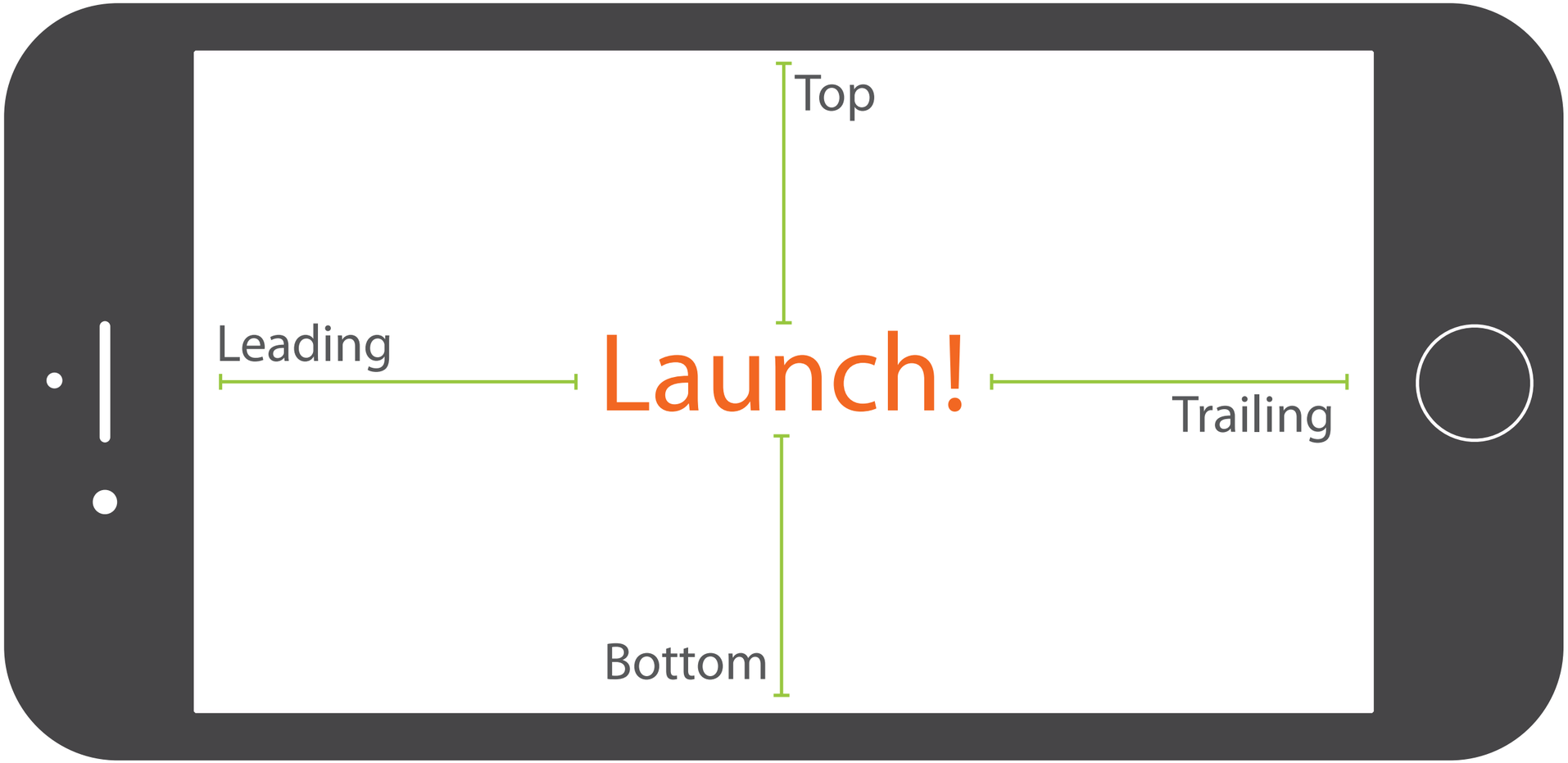
James Wilson

www.noesisingenuity.com | @jam_wils

Module Overview

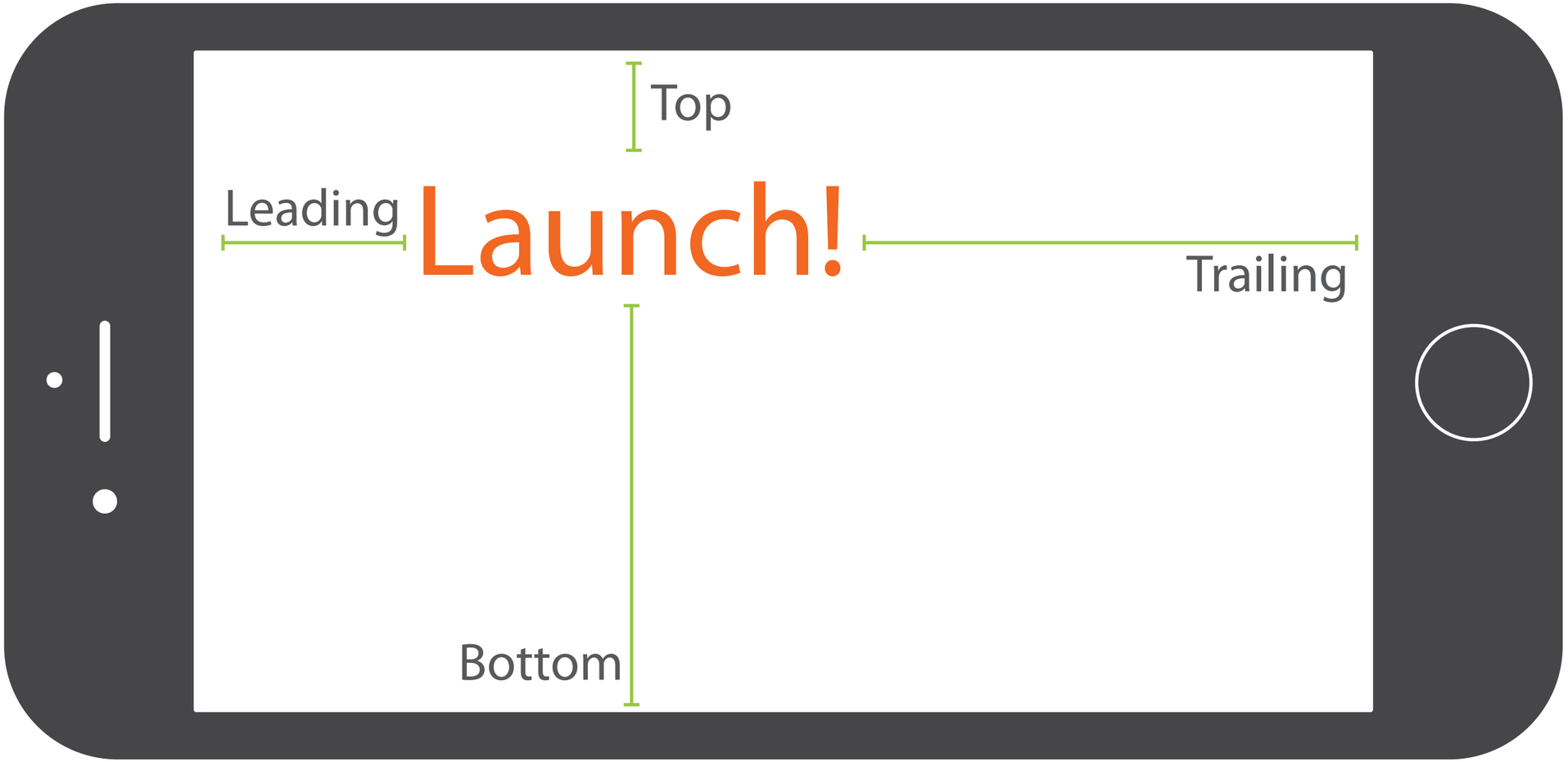
- Common Constraints
- Key Components
- How to Build Constraints
- Satisfying the Constraints
- Linear Equations
- Constraint Demo

The Constraints

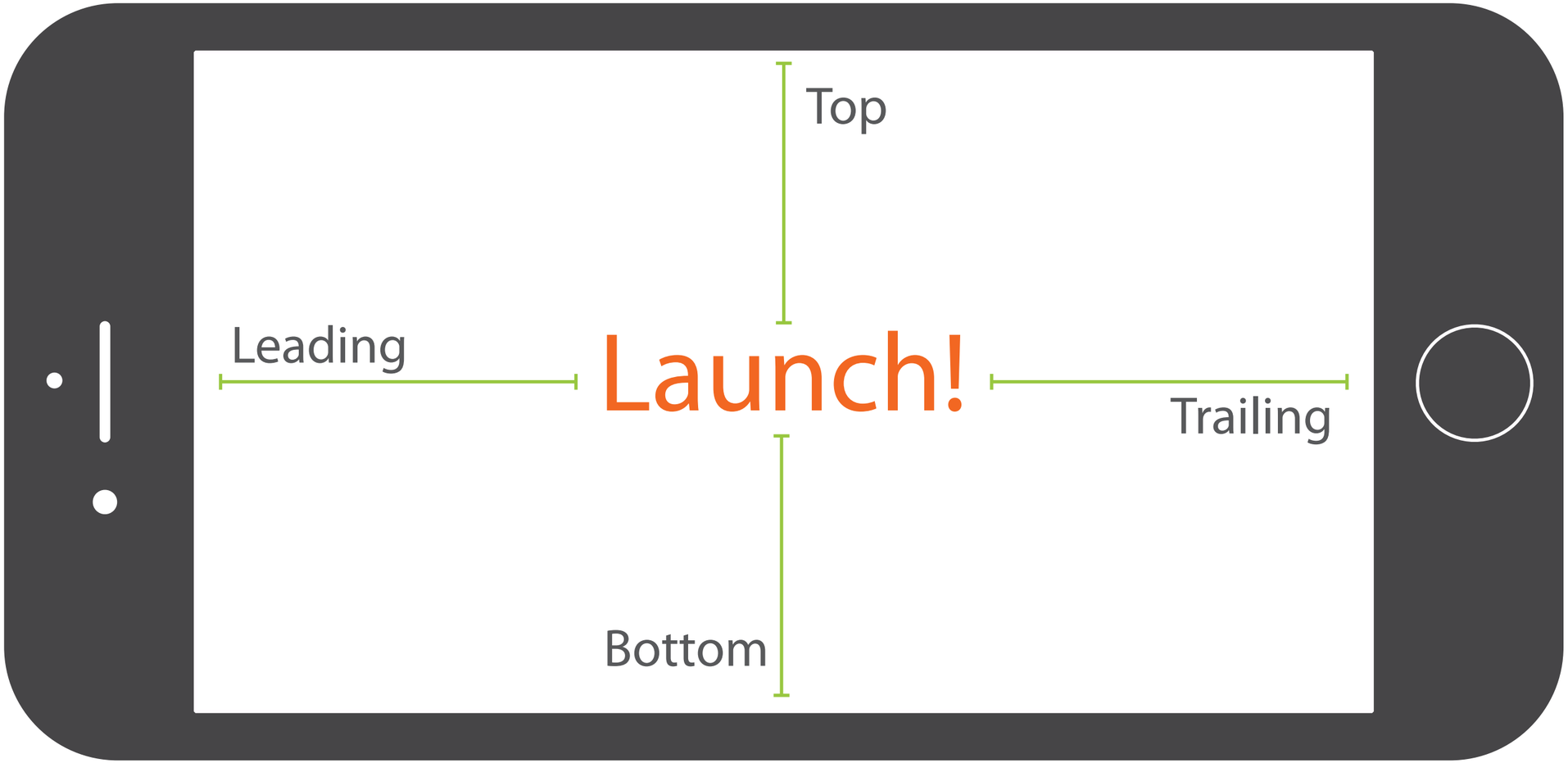


The Constraints

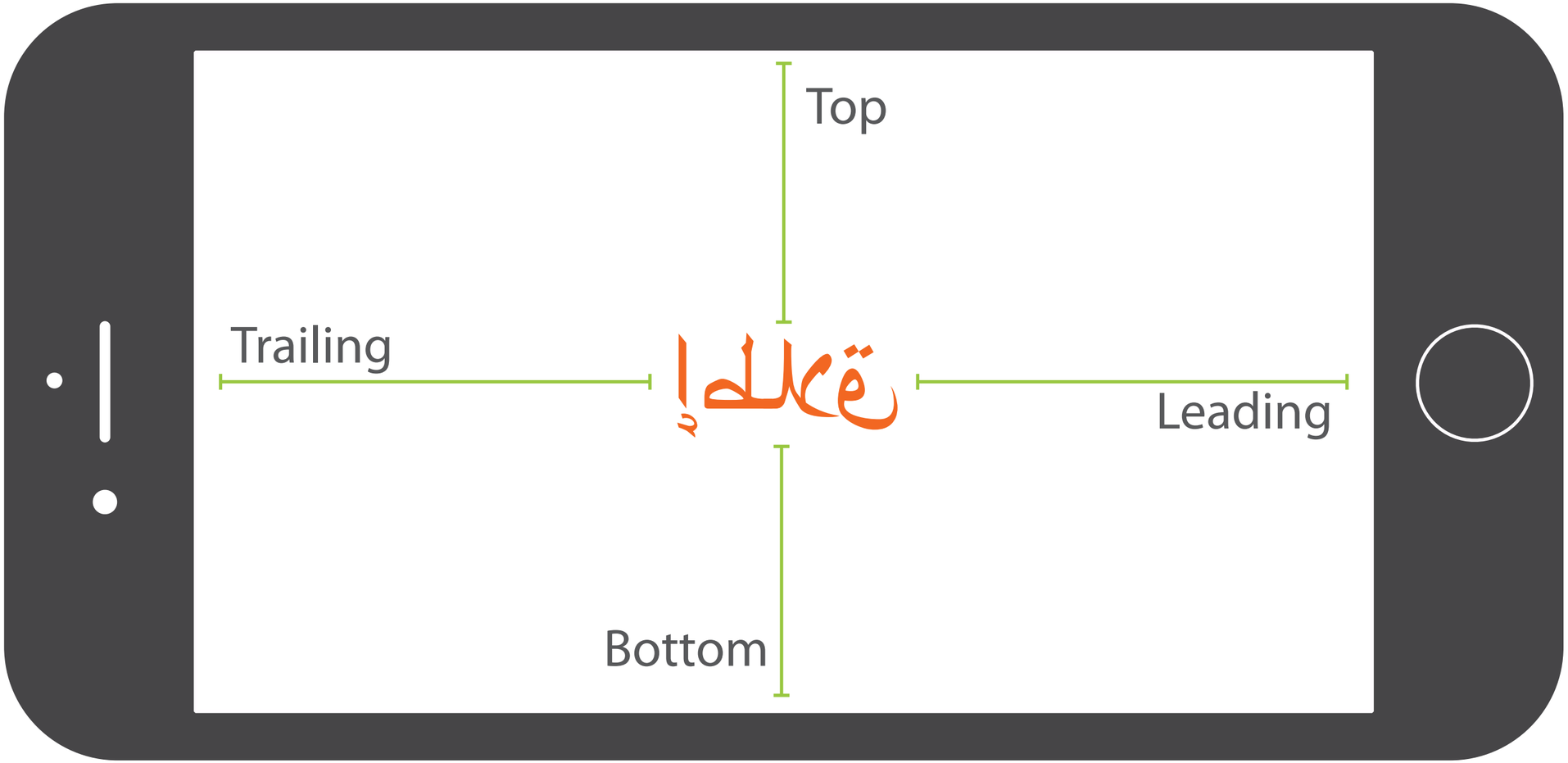
The Constraints



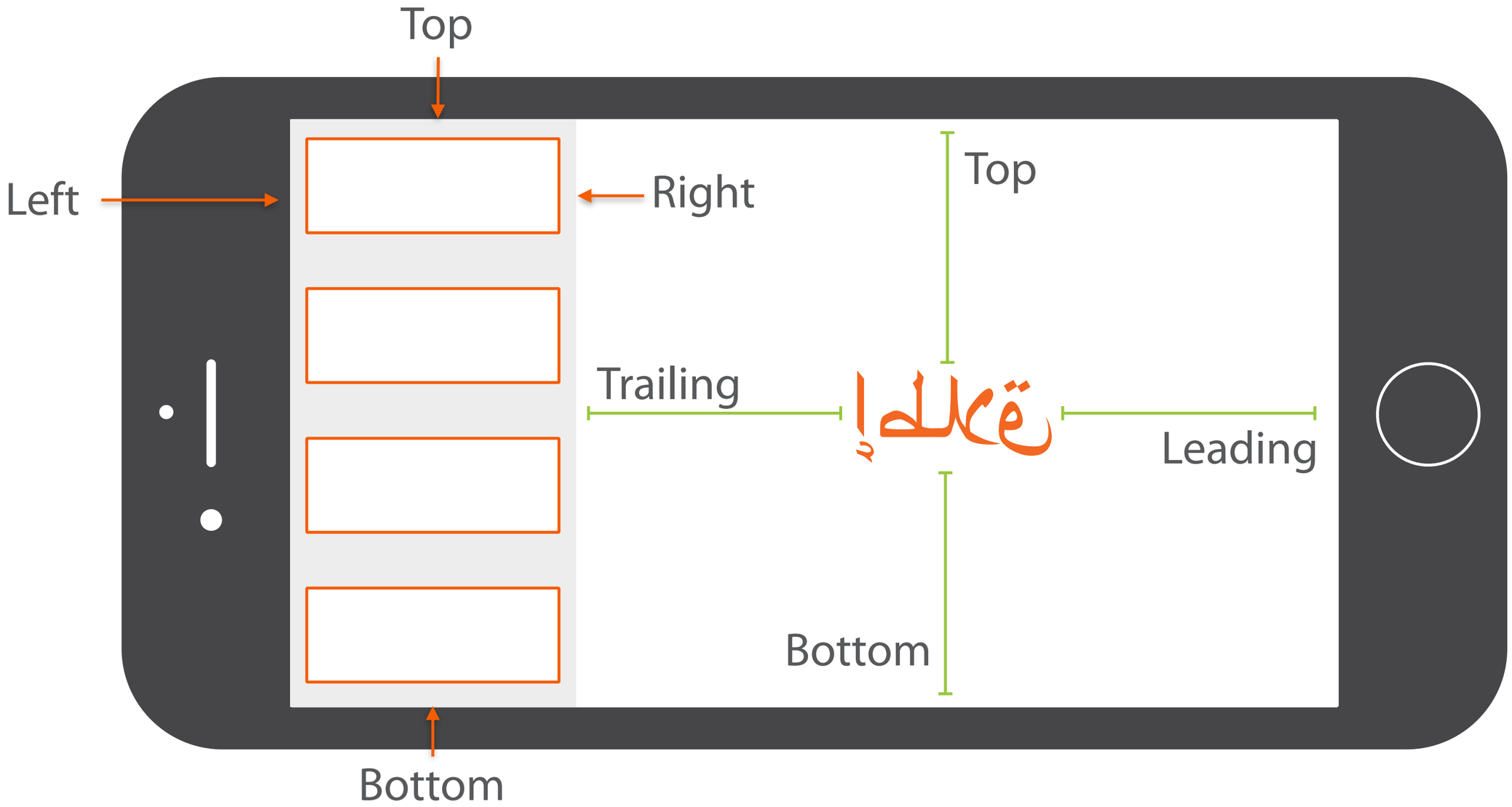
The Constraints



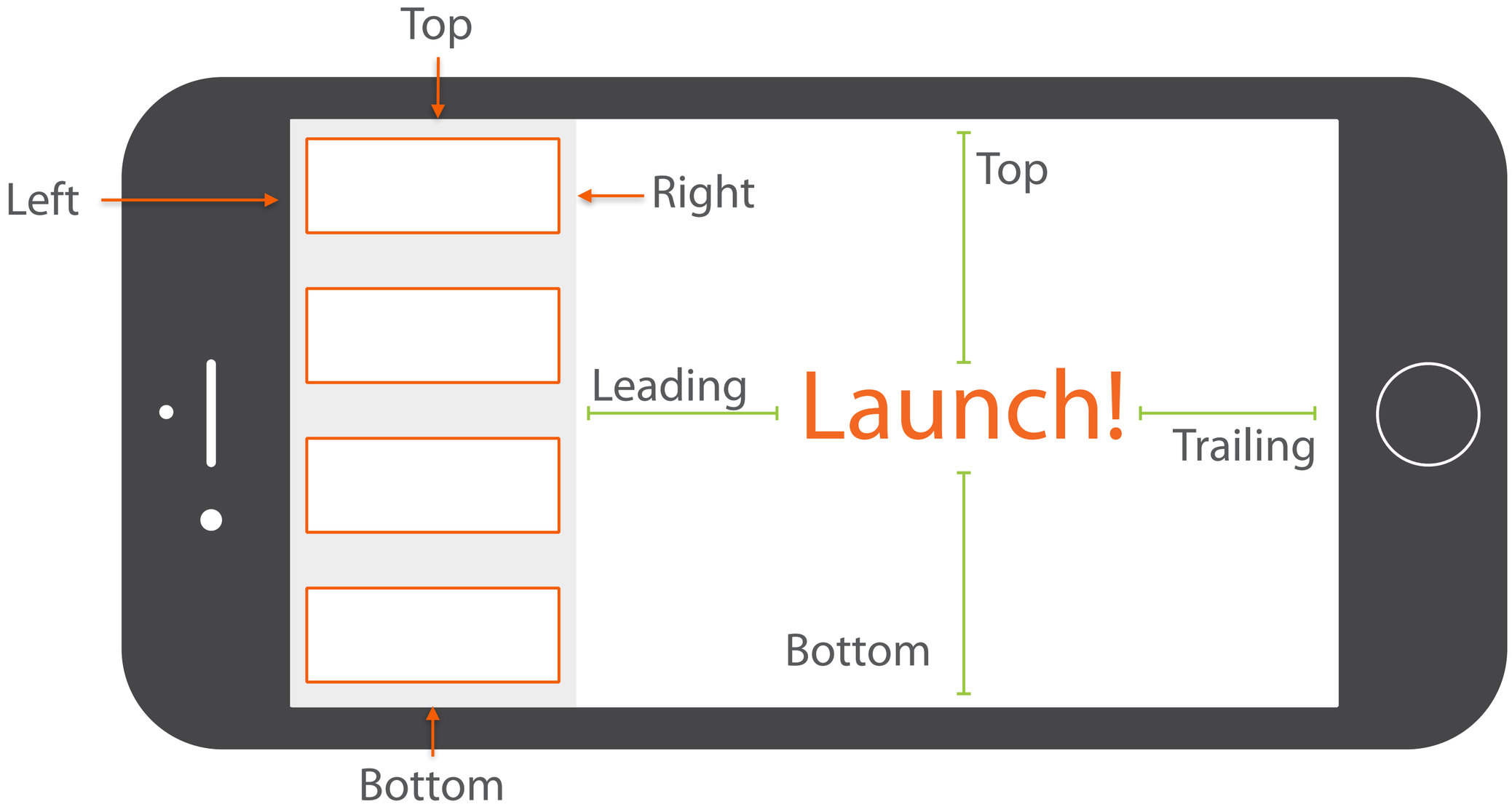
The Constraints



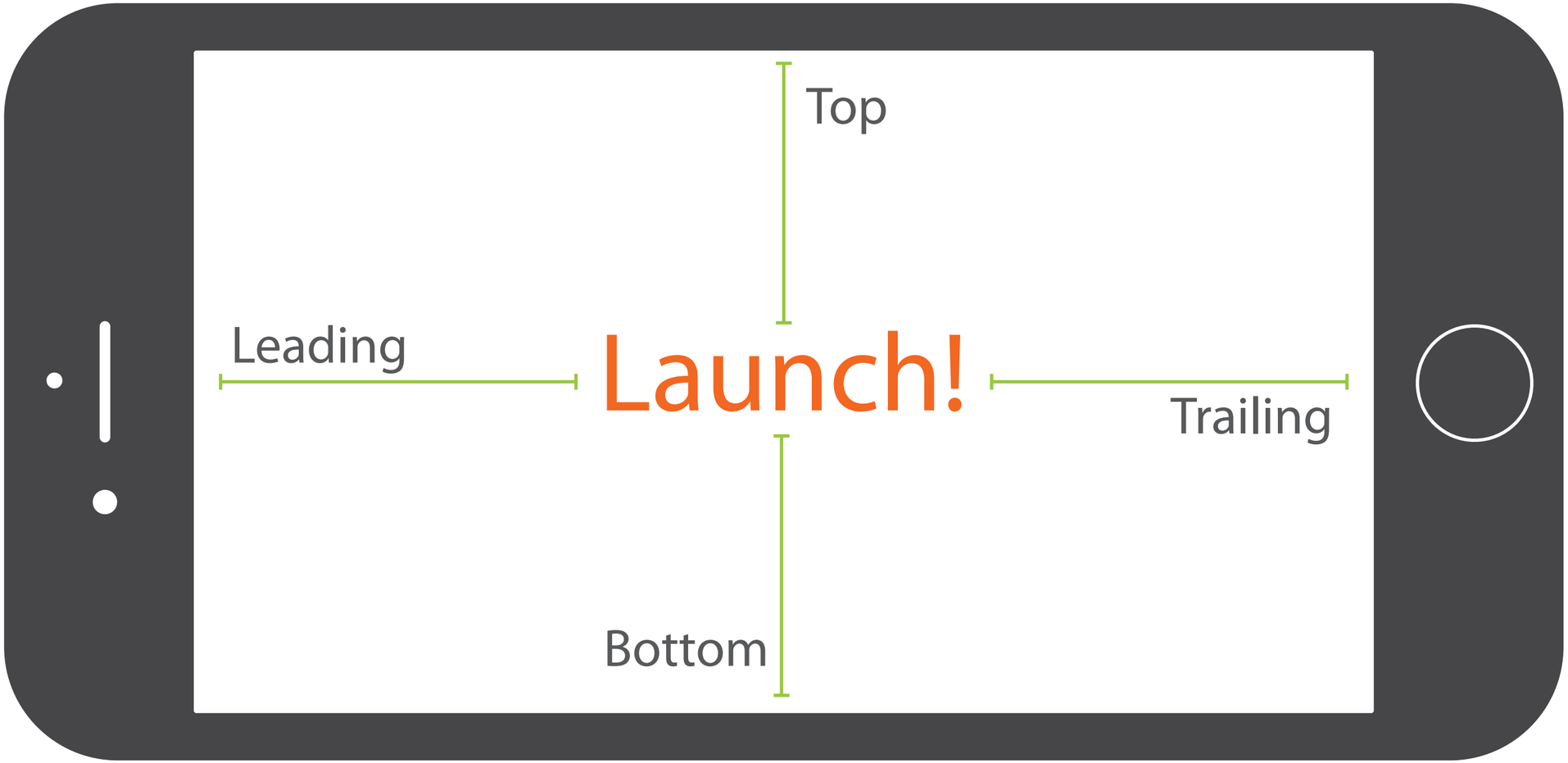
The Constraints



The Constraints

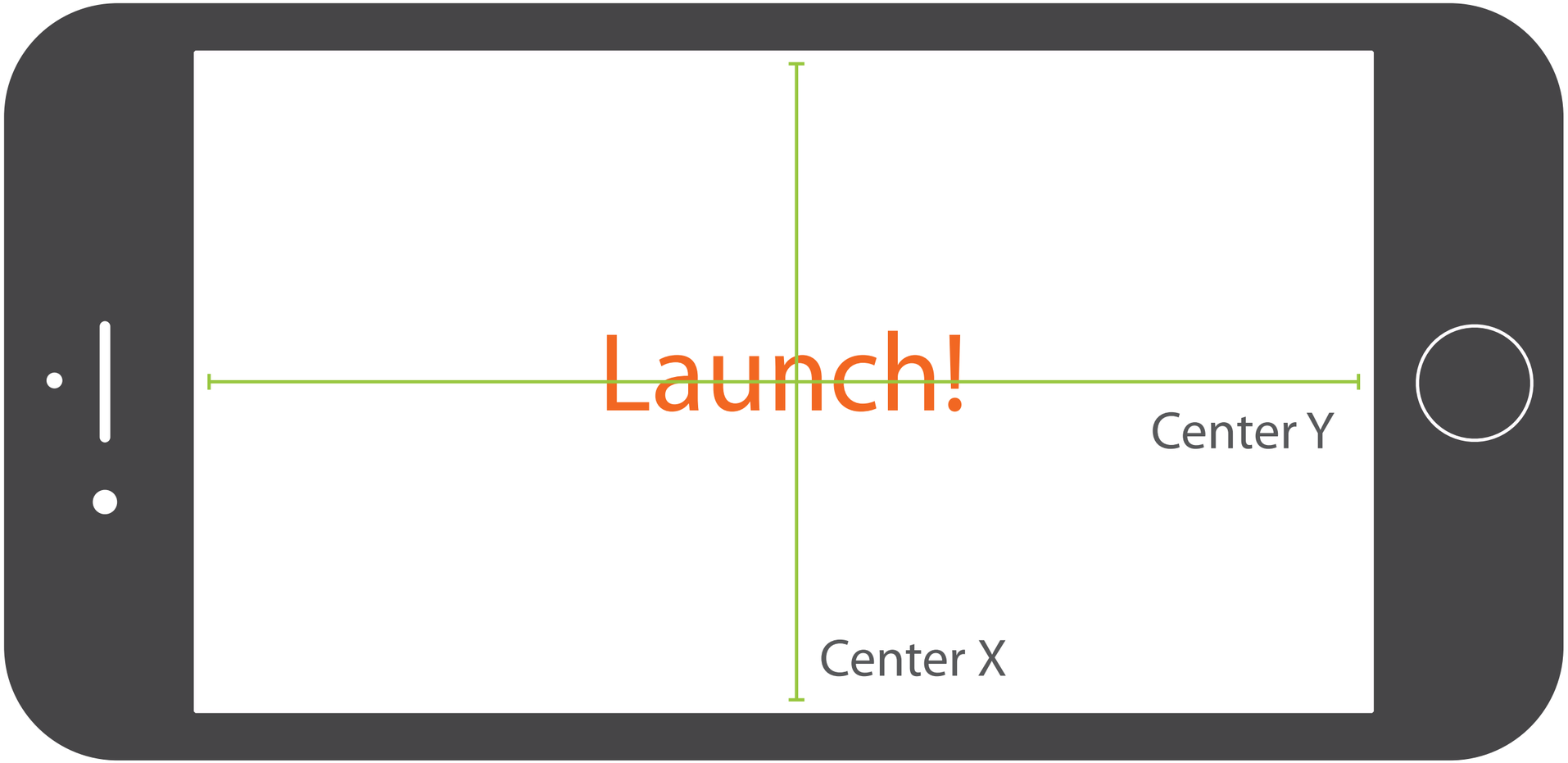


The Constraints

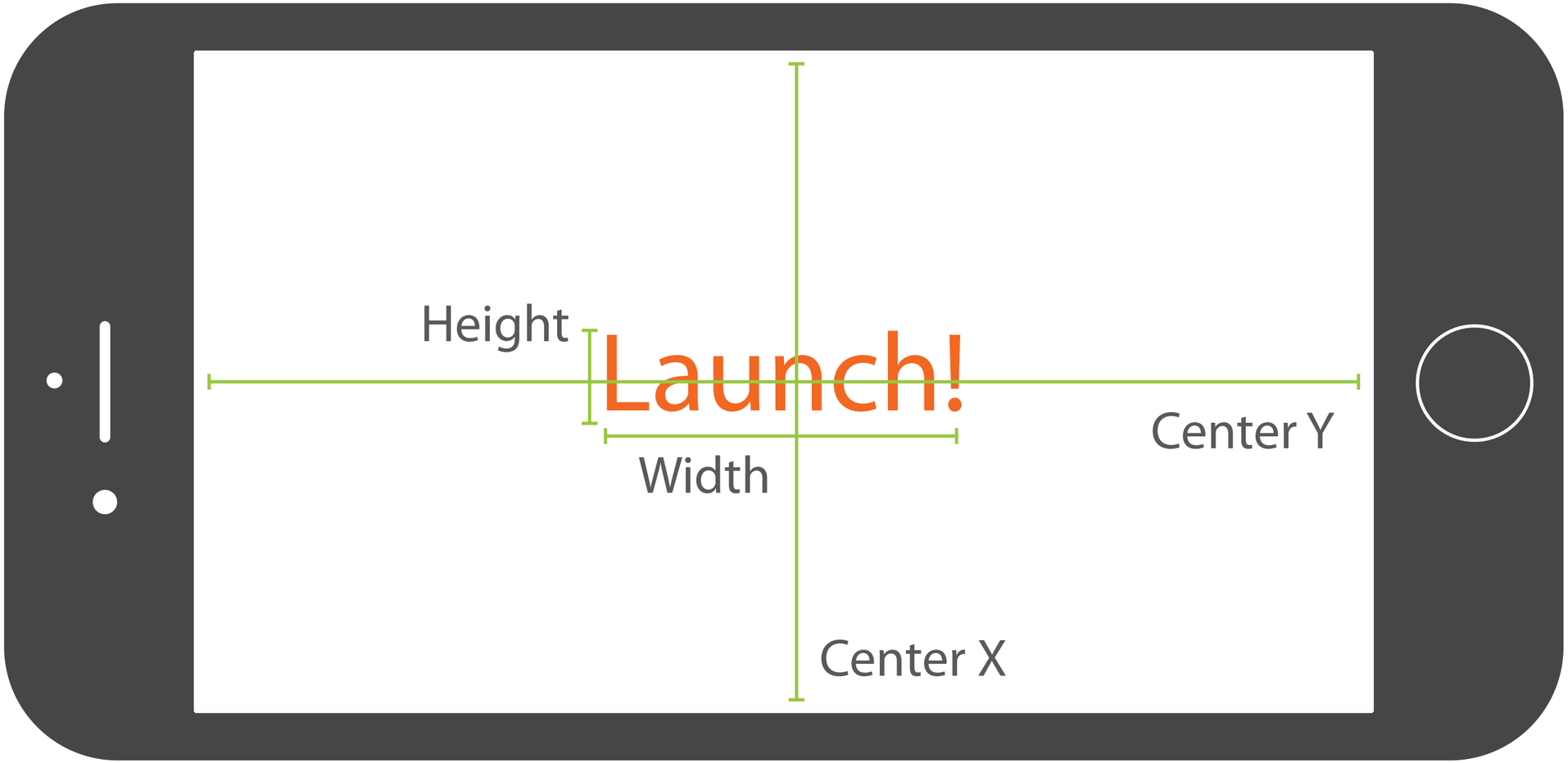


The Constraints

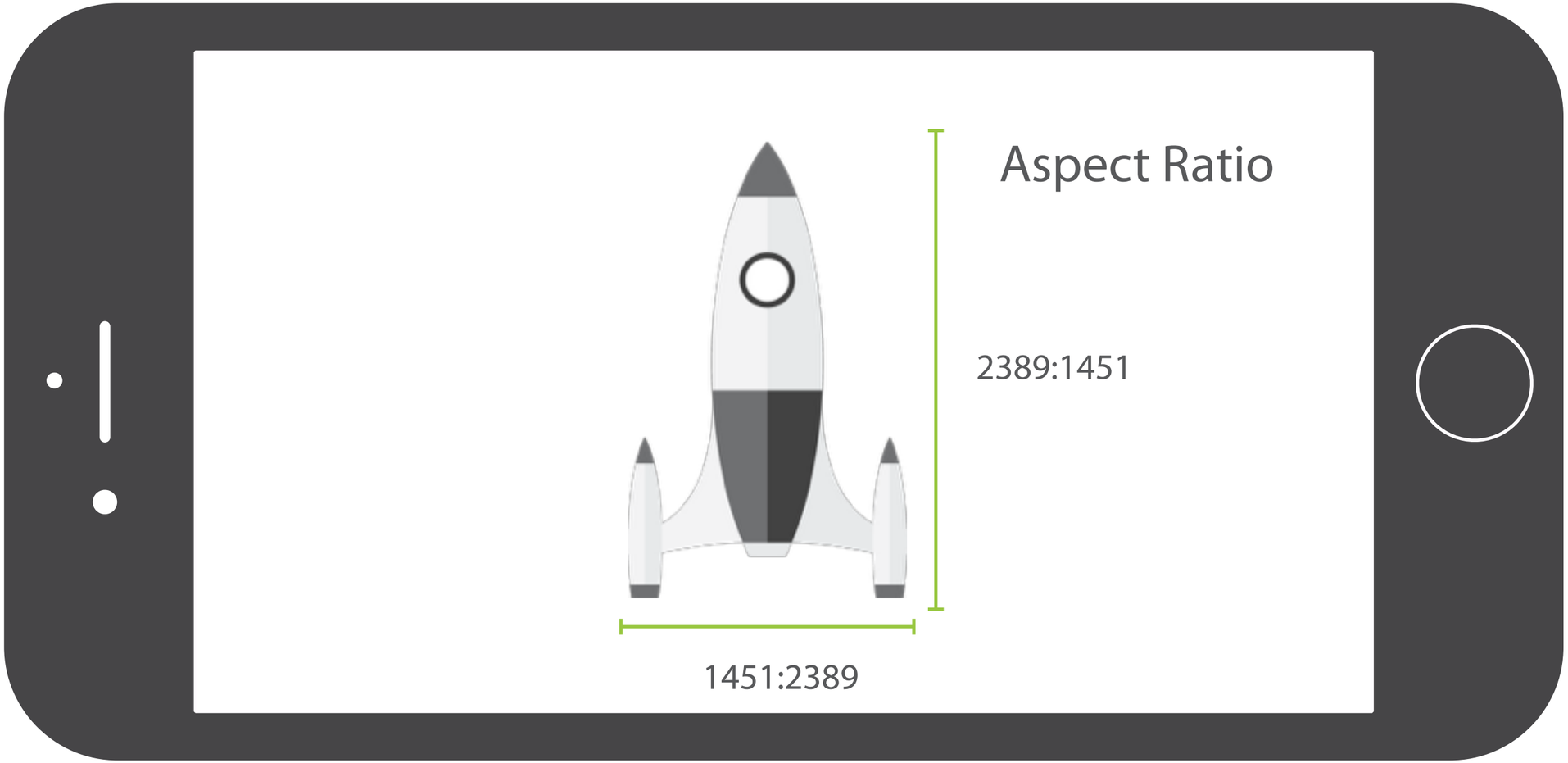
The Constraints



The Constraints

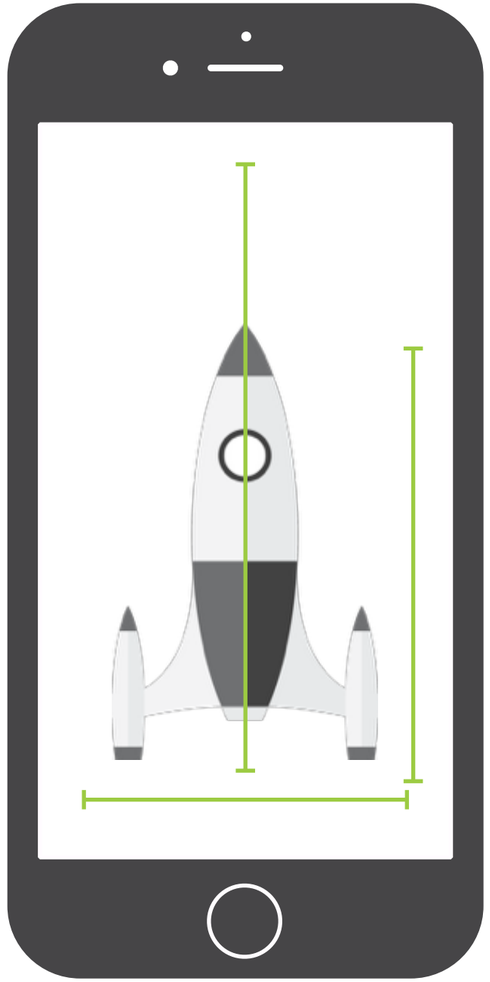


The Constraints



The Constraints

The Key Components



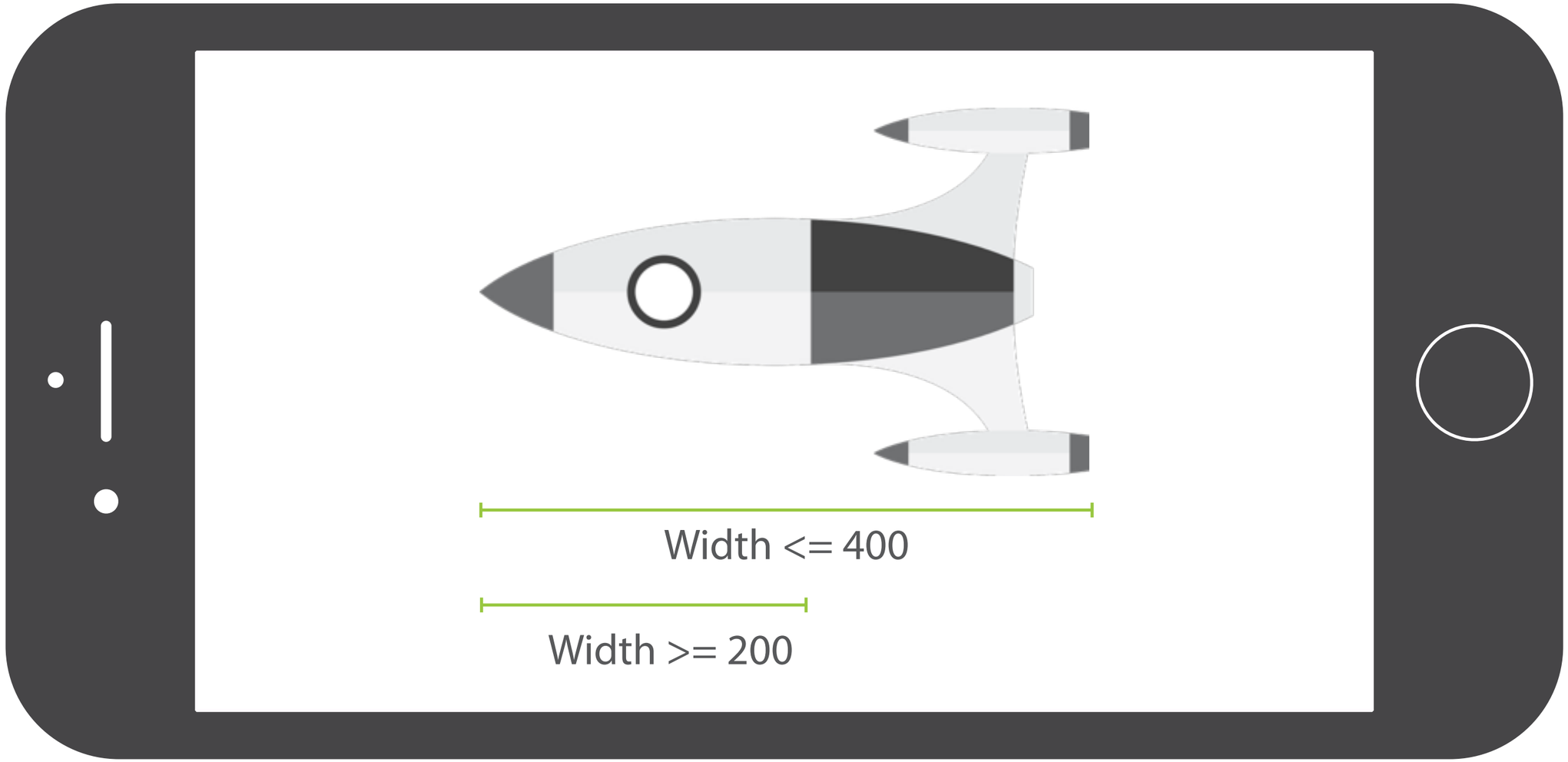
Constant

Number that defines the distance

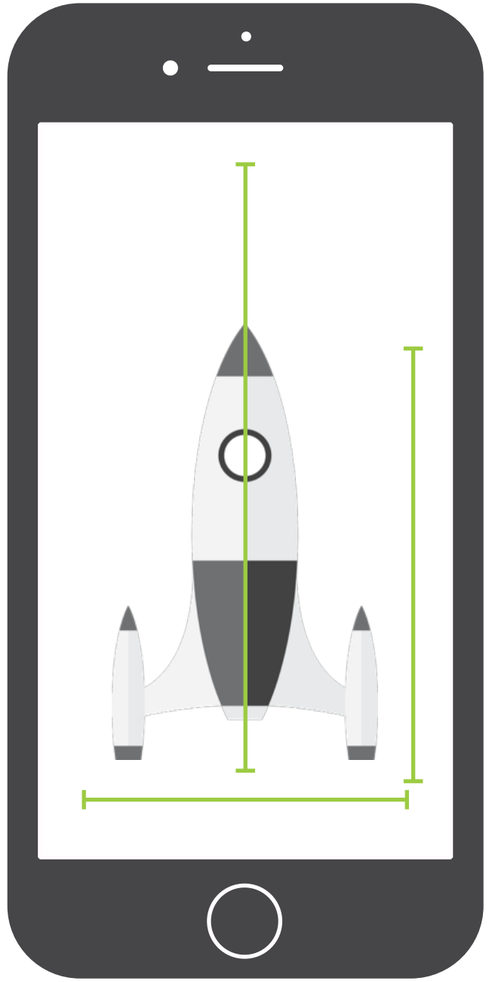
Relation

= >= <=

Relation



The Key Components



Constant

Number that defines the distance

Relation

= >= <=

Priority

Zero is the lowest priority, and 1000 is required

Building Constraints

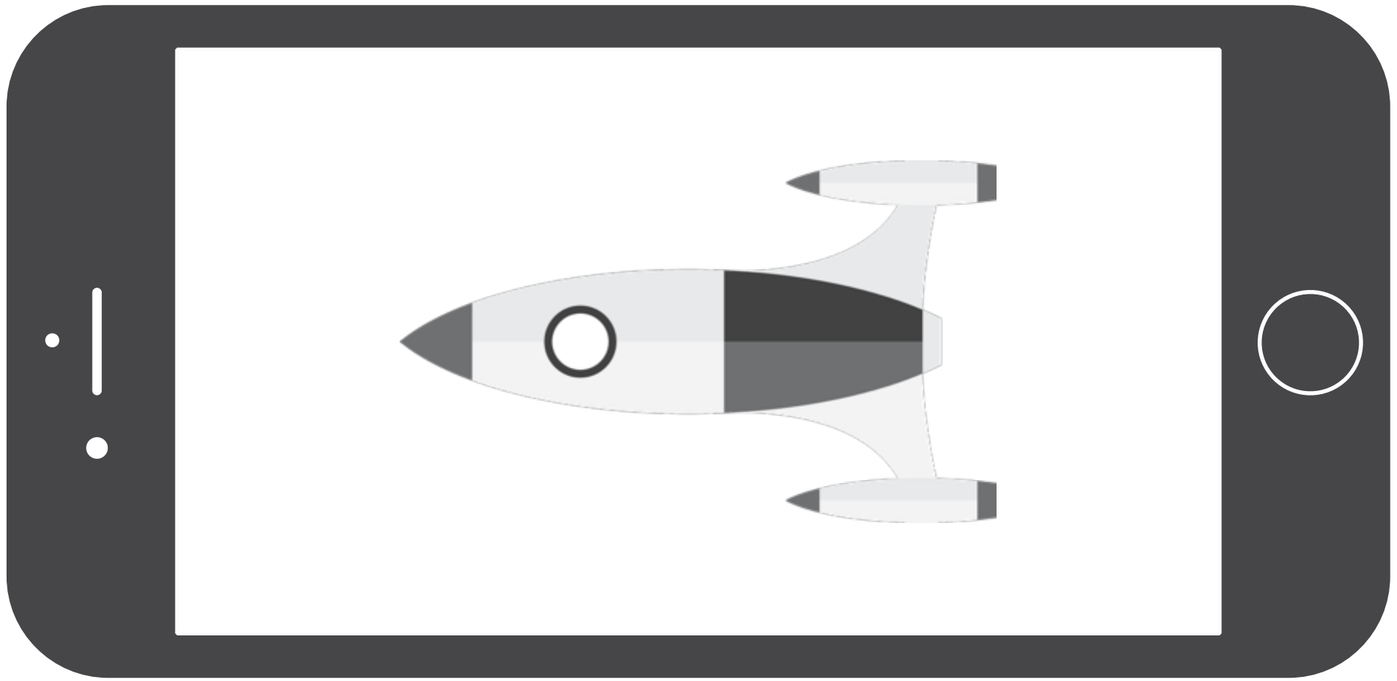
Interface Builder | Visual Format Language | NSLayoutConstraint API

V:|-(40@600)-[item1]-[item2(==item1)]-20-[item3(==item1)]-[item4(==item3)]-(>=40)-|

Your Mission

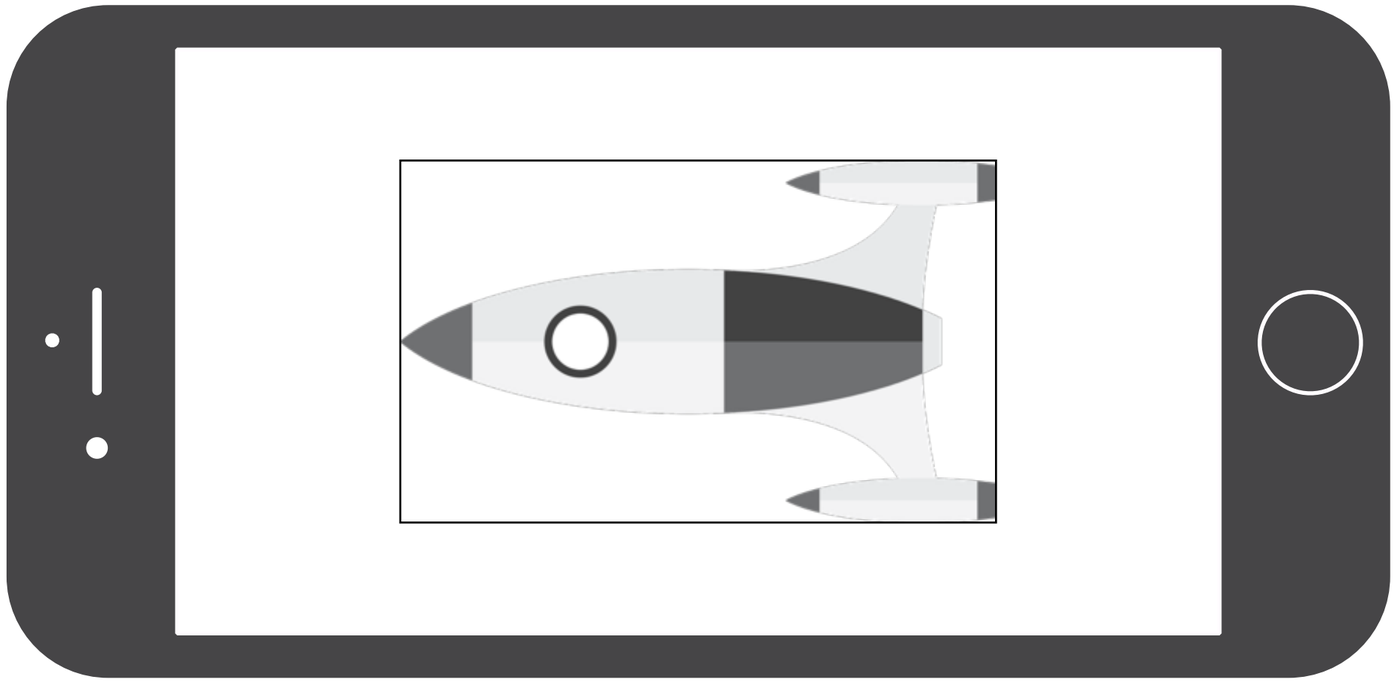
Finish the UI Design | Satisfy the Constraints |

Satisfy the Constraints



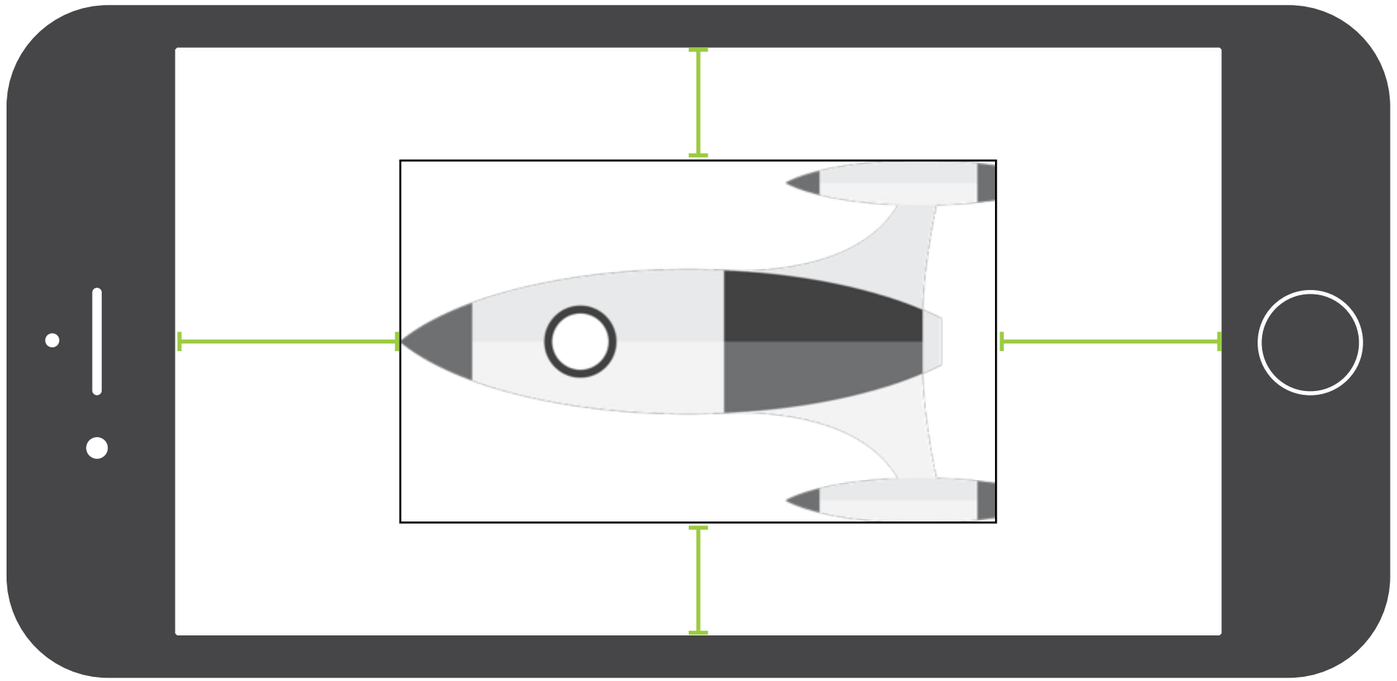
- X-Coordinate
- Y-Coordinate
- Width
- Height

Satisfy the Constraints



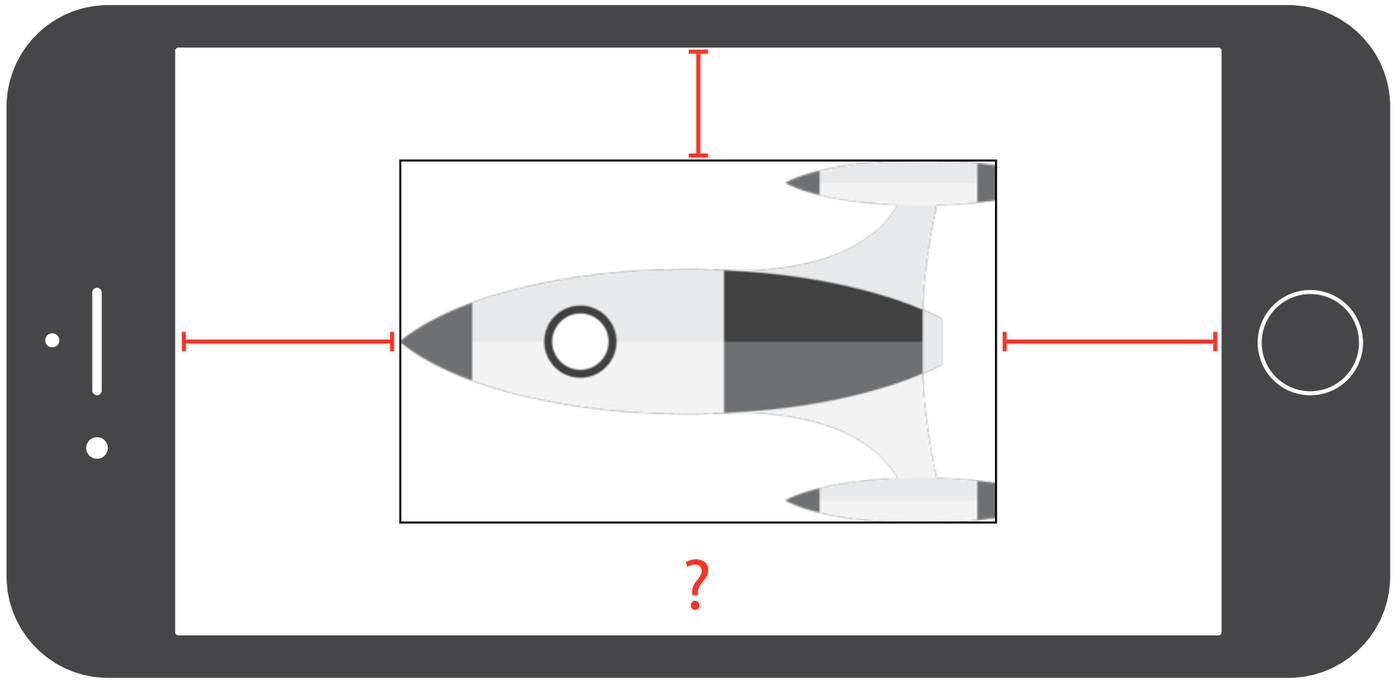
- X-Coordinate
- Y-Coordinate
- Width
- Height

Satisfy the Constraints



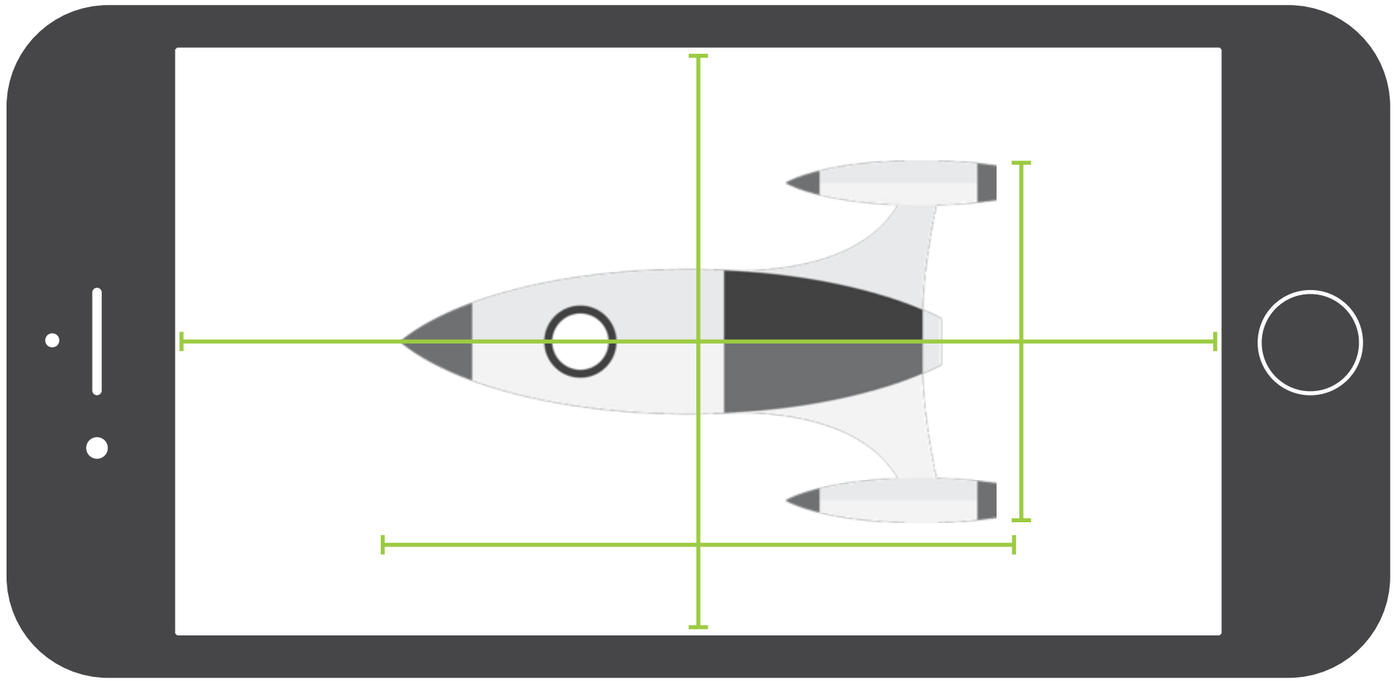
- X-Coordinate
- Y-Coordinate
- Width
- Height

Satisfy the Constraints



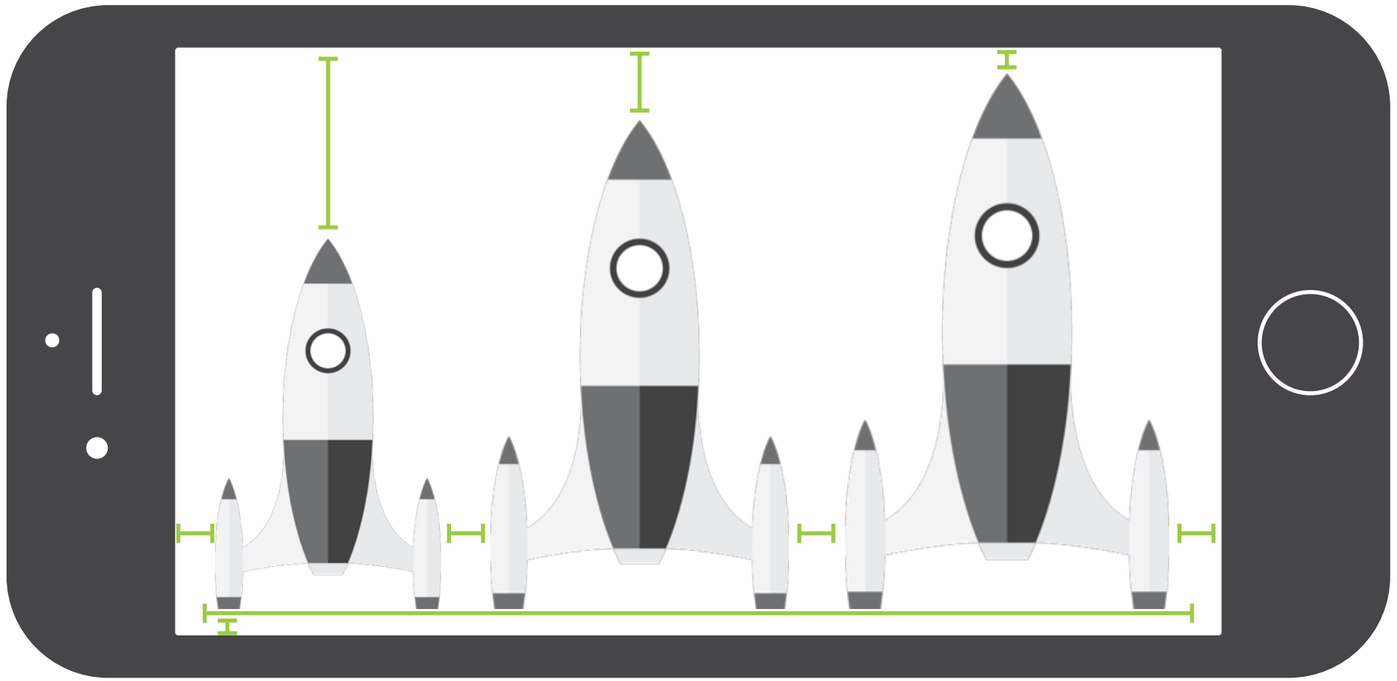
- X-Coordinate
- Y-Coordinate
- Width
- Height

Satisfy the Constraints



- X-Coordinate
- Y-Coordinate
- Width
- Height

Satisfy the Constraints



- X-Coordinate
- Y-Coordinate
- Width
- Height

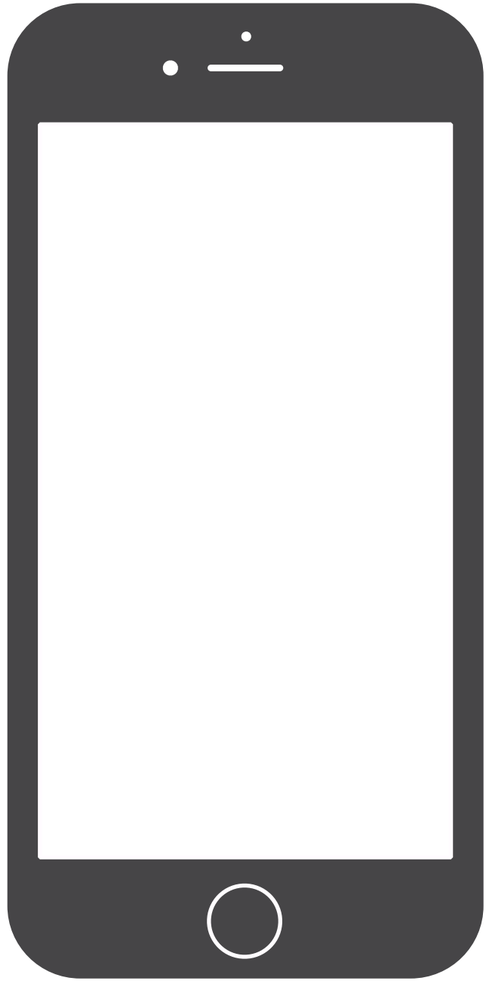
Your Mission

Finish the UI Design | Satisfy the Constraints | Think About Relationships

The Linear Equation

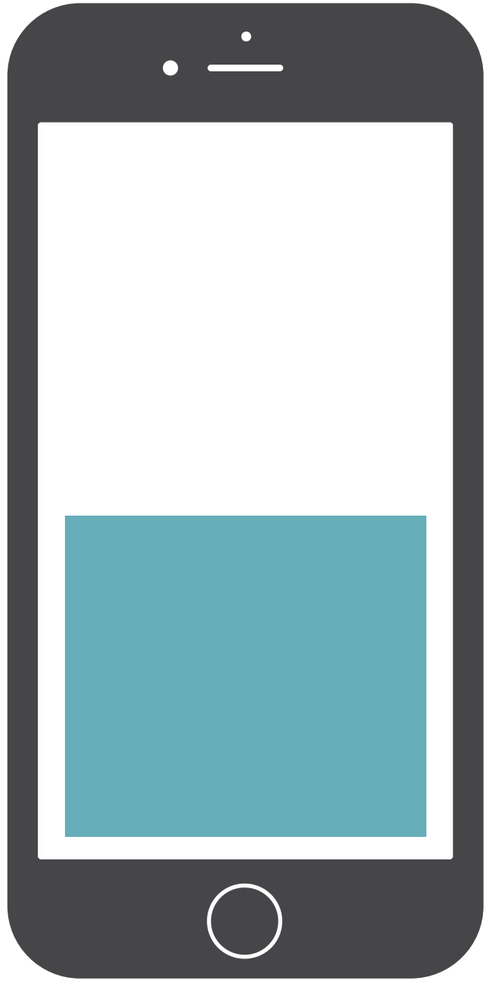
$$y = mx + b$$

The Linear Equation



$$y = m \cdot x + b$$

The Linear Equation

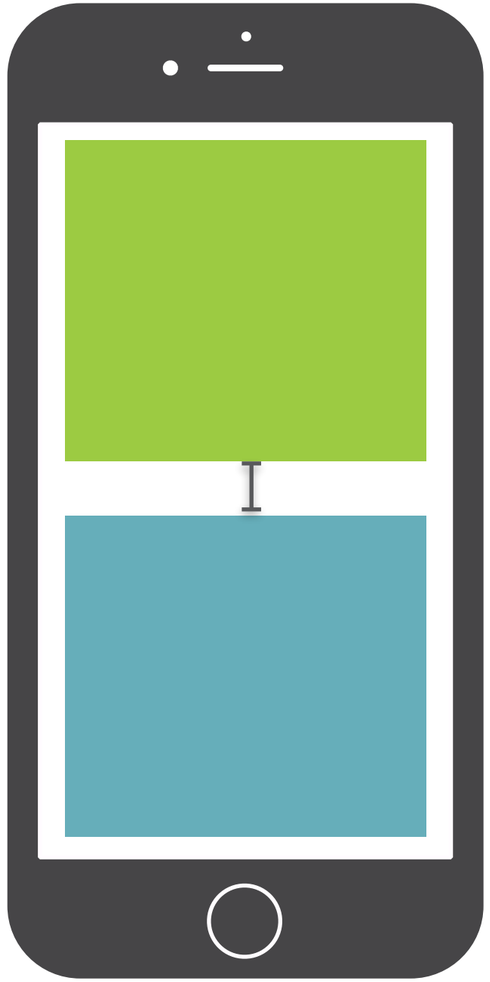


$$\text{BlueView.Top} = m \cdot x + b$$

The Linear Equation

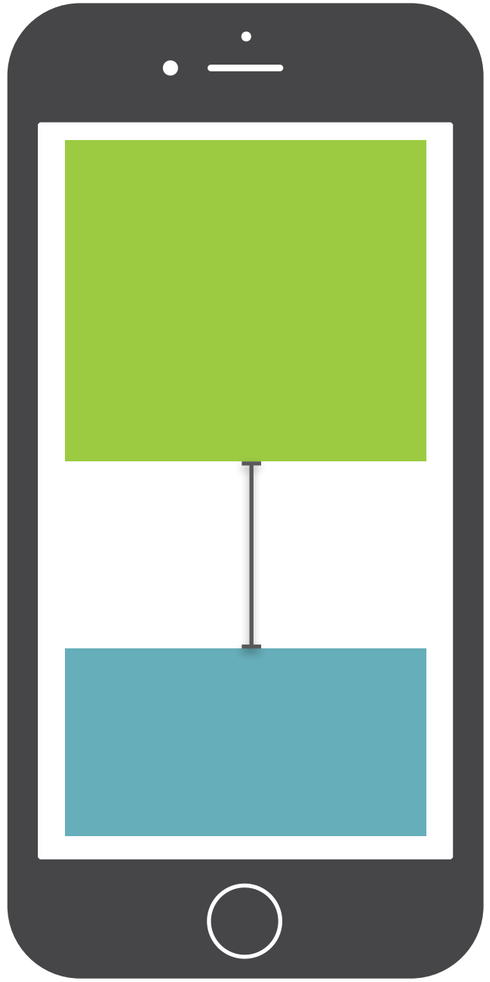
$$BlueView.Top = m \cdot GreenView.Bottom + b$$

The Linear Equation



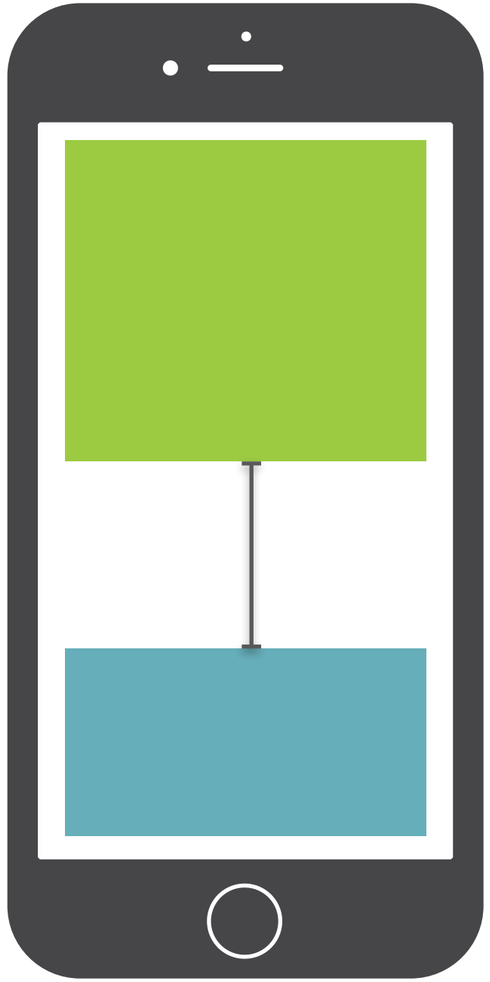
$$BlueView.Top = 1 \cdot GreenView.Bottom + 10$$

The Linear Equation



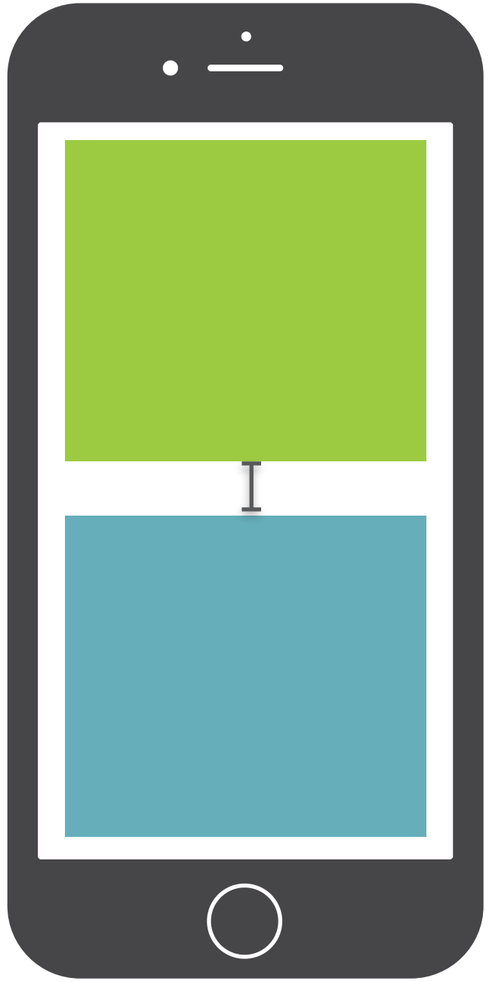
$$BlueView.Top = 1 \cdot GreenView.Bottom + 40$$

The Linear Equation



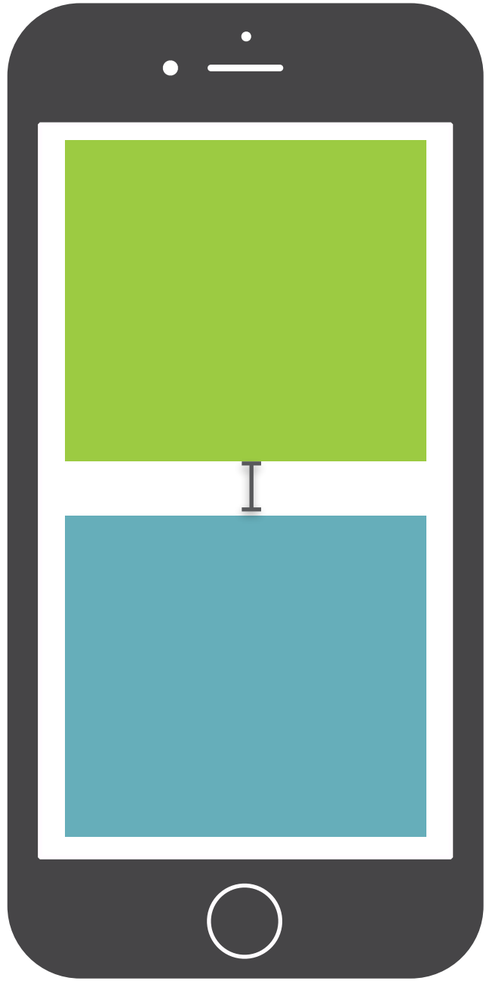
$$BlueView.Top = 2 \cdot GreenView.Bottom + 20$$

The Linear Equation



$$BlueView.Top = 2 \cdot GreenView.Bottom + 5$$

The Linear Equation



$$BlueView.Top = 2 \cdot GreenView.Bottom + 5$$

```
let constraint = NSLayoutConstraint(
  item: AnyObject,
  attribute: NSLayoutAttribute,
  relatedBy: NSLayoutRelation,
  toItem: AnyObject?,
  attribute: NSLayoutAttribute,
  multiplier: CGFloat,
  constant: CGFloat
)
```

$$\text{BlueView.Top} = 2 \cdot \text{GreenView.Bottom} + 5$$

```
let constraint = NSLayoutConstraint(
  item: AnyObject,
  attribute: NSLayoutAttribute,
  relatedBy: NSLayoutRelation,
  toItem: AnyObject?,
  attribute: NSLayoutAttribute,
  multiplier: CGFloat,
  constant: CGFloat
)
```

$$\text{BlueView.Top} = 2 \cdot \text{GreenView.Bottom} + 5$$

```
let constraint = NSLayoutConstraint(
   item: BlueView,
   attribute: NSLayoutAttribute,
   relatedBy: NSLayoutRelation,
   toItem: AnyObject?,
   attribute: NSLayoutAttribute,
   multiplier: CGFloat,
   constant: CGFloat
)
```

.Top = 2 · GreenView.Bottom + 5

```
let constraint = NSLayoutConstraint(
   item: BlueView,
   attribute: NSLayoutAttribute.Top,
   relatedBy: NSLayoutRelation,
   toItem: AnyObject?,
   attribute: NSLayoutAttribute,
   multiplier: CGFloat,
   constant: CGFloat
)
```

$$= 2 \cdot \text{GreenView.Bottom} + 5$$

```
let constraint = NSLayoutConstraint(
   item: BlueView,
   attribute: NSLayoutAttribute.Top,
   relatedBy: NSLayoutRelation.Equal,
   toItem: AnyObject?,
   attribute: NSLayoutAttribute,
   multiplier: CGFloat,
   constant: CGFloat
)
```

$$2 \cdot \text{GreenView.Bottom} + 5$$

```
let constraint = NSLayoutConstraint(
   item: BlueView,
   attribute: NSLayoutAttribute.Top,
   relatedBy: NSLayoutRelation.Equal,
   toItem: GreenView,
   attribute: NSLayoutAttribute,
   multiplier: CGFloat,
   constant: CGFloat
)
```

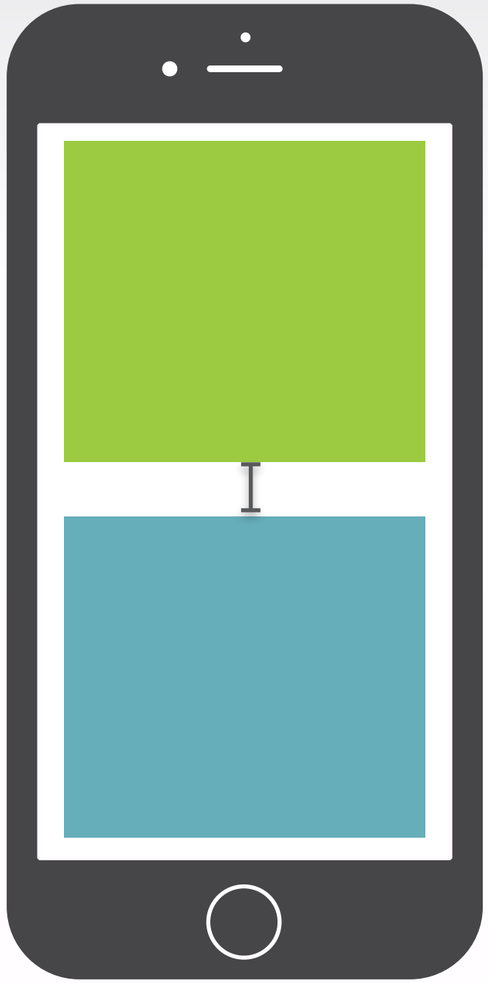
```
2 ·                    .Bottom + 5
```

```
let constraint = NSLayoutConstraint(
   item: BlueView,
   attribute: NSLayoutAttribute.Top,
   relatedBy: NSLayoutRelation.Equal,
   toItem: GreenView,
   attribute: NSLayoutAttribute.Bottom,
   multiplier: CGFloat,
   constant: CGFloat
)
```

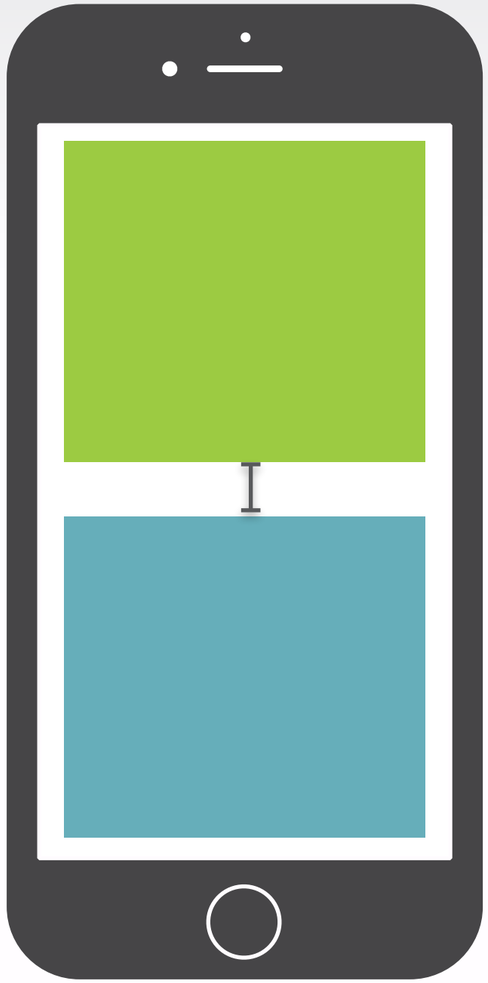
$$2 \cdot \qquad + 5$$

```
let constraint = NSLayoutConstraint(
   item: BlueView,
   attribute: NSLayoutAttribute.Top,
   relatedBy: NSLayoutRelation.Equal,
   toItem: GreenView,
   attribute: NSLayoutAttribute.Bottom,
   multiplier: 2,
   constant: 5
)
```

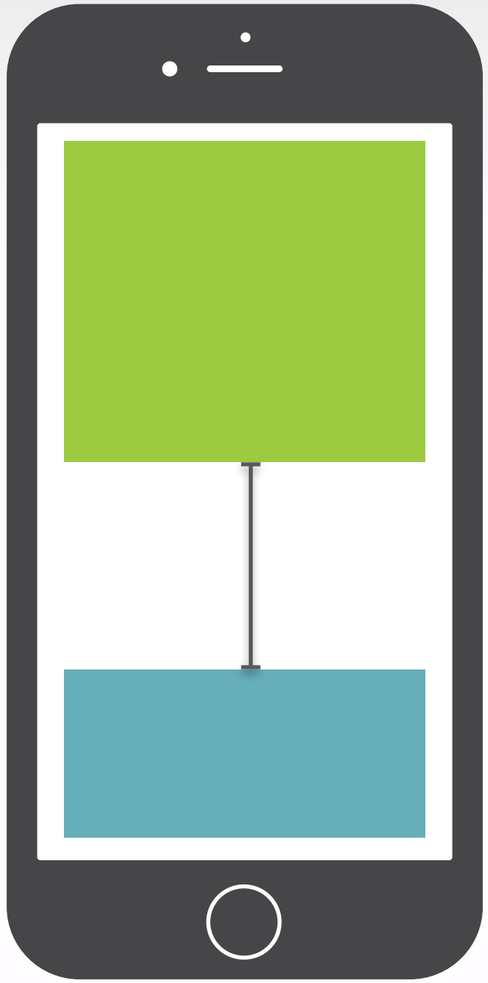
```
let constraint = NSLayoutConstraint(
   item: BlueView,
   attribute: NSLayoutAttribute.Top,
   relatedBy: NSLayoutRelation.Equal,
   toItem: GreenView,
   attribute: NSLayoutAttribute.Bottom,
   multiplier: 2,
   constant: 5
)
```



```
let constraint = NSLayoutConstraint(
    item: BlueView,
    attribute: NSLayoutAttribute.Top,
    relatedBy: NSLayoutRelation.Equal,
    toItem: GreenView,
    attribute: NSLayoutAttribute.Bottom,
    multiplier: 2,
    constant: 5
)
```



```
let constraint = NSLayoutConstraint(
    item: BlueView,
    attribute: NSLayoutAttribute.Top,
    relatedBy: NSLayoutRelation.Equal,
    toItem: GreenView,
    attribute: NSLayoutAttribute.Bottom,
    multiplier: 2,
    constant: 5
)
```



```
let constraint = NSLayoutConstraint(
    item: BlueView,
    attribute: NSLayoutAttribute.Top,
    relatedBy: NSLayoutRelation.Equal,
    toItem: GreenView,
    attribute: NSLayoutAttribute.Bottom,
    multiplier: 1,
    constant: 40
)
```

Summary

Various Constraints

Constants, Priorities, and Relations

Satisfying the Constraints

Constraint's Linear Equation

Constraint Demo