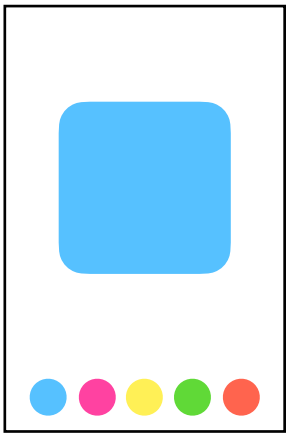
Coding Test document

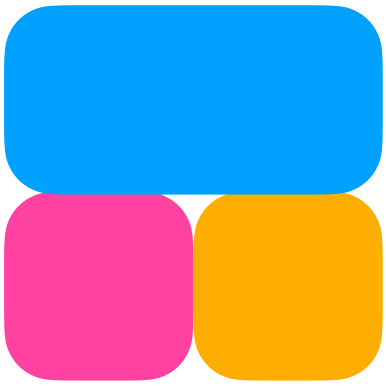
Screen



```
Vstack {
    Widget View
    HStack {
        Color circle items
    }
}
```

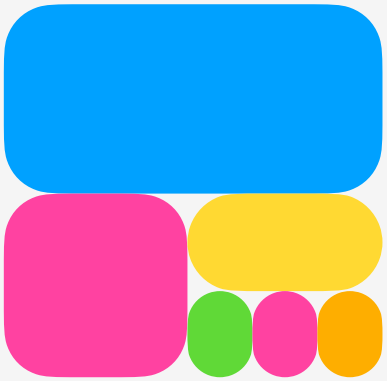
Widget View

Combination of:
- A StackView-behavior-view
- Color Item



```
Vstack {
    Blue Item

    HStack {
        Pink item
        Yellow item
    }
}
```



```
Vstack {
    Blue Item

    HStack {
        Pink item

        Vstack {
            Yellow Item
            HStack {
                Green Item
                Pink item
                Orange item
            }
        }
    }
}
```

Data type

a. Stack

```
class Stack: Item {
    let id = UUID()
    var contents: [Item]
    var axis: Axis
    var rect: CGRect {
        didSet {
            adjustContentsSize()
        }
    }
}
```

- ***id*** to check it identity, easier to compare to other stack, find item.
- ***contents*** to store items.
- ***axis*** of this stack: Vertical or Horizontal.
- ***rect***, when it adjust size, content inside it also adjust to appropriate sizes.

b. Color Item

```
class ColorItem: Item {
    let id = UUID()
    var rect: CGRect = .zero
    var color: Color = .clear
}
```

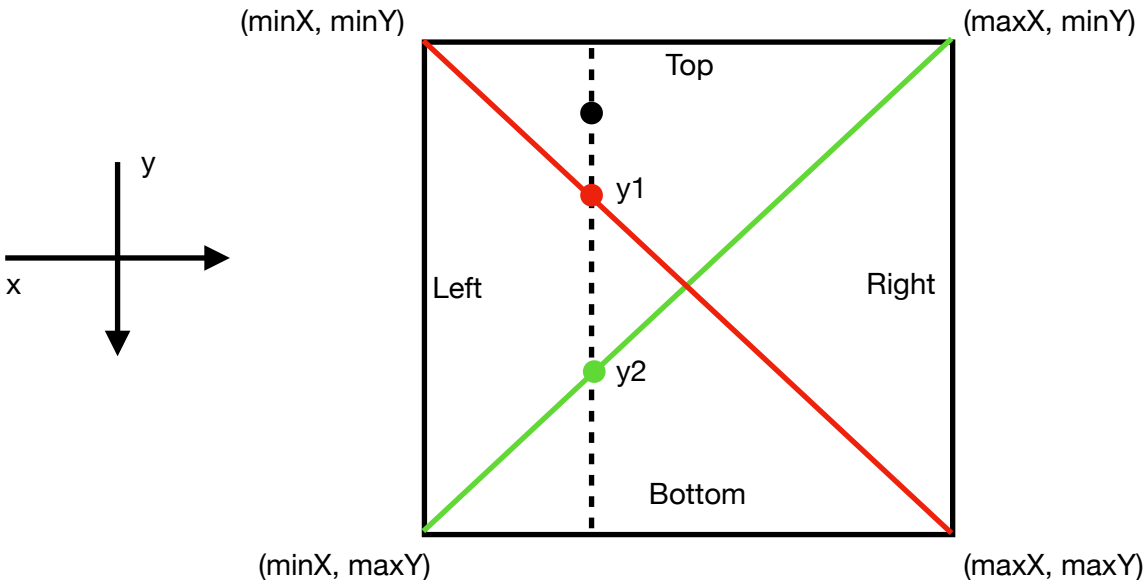
- ***id*** to check it identity, easier to compare to other item, find item.
- ***color***, color of this item.

c. Item protocol

```
protocol Item {
    var id: UUID { get }
    var rect: CGRect { get set }
}
```

- ***Stack*** and ***Color Item*** conform to this protocol to be able to stay in an array together.

Hit Test



- The purpose is to identify which zone (top, left, right, bottom) does the black dot belong to.
- Given a random point in square, we can always find a reflection if this point on 2 diagonals (red line, green line).
- Compare y of black dot to y of red dot & y of green dot to determine.

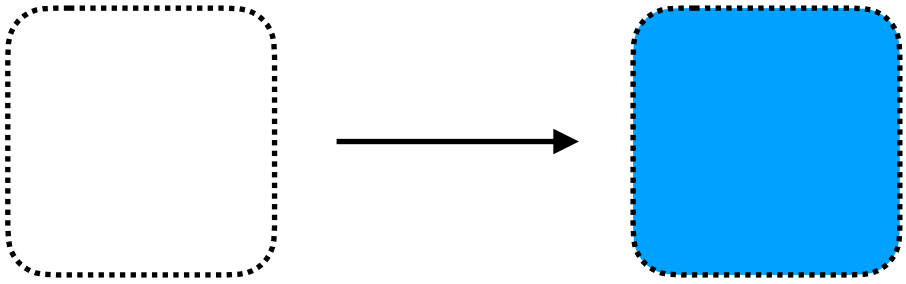
Medium	$m = \frac{maxY - minY}{maxX - minX}$
y of red dot	**y1 = x * m + (minY - m * minX)**
y of green dot	**y2 = maxY - y1 + minY**

- Logic

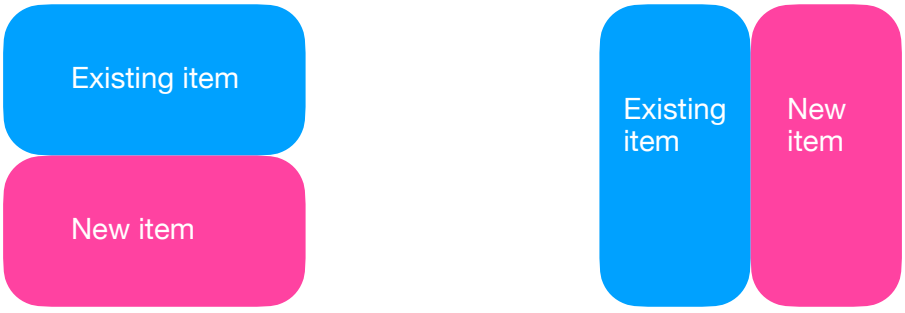
Condition	Zone	Axis
y < y1 & y < y2	Top	Vertical
y < y1 & y >= y2	Right	Horizontal
y >= y1 & y < y2	Left	Horizontal
y >=y1 & y >= y2	Bottom	Vertical

Adding item

Default View has a Stack in it already, adding 1 color item will take the whole space.

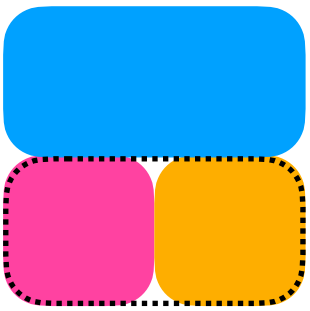


Adding a new color item, the existing item will make a space for the new item, then change it space based on hit test, stack also change its axis.



Adding new item that has different axis with the current stack, will create a new stack with current touching item and new item.

- Remove touching item in old stack.
- Adding new stack with old item & new item.



The end.