



iOS Crew

15– 19 апреля

Хардкорный UI

Александр Филимонов

SmileCraft

Мастер-класс.

UICollectionViewLayout:

от каменного века

до Compositional Layout



О себе

- 7 лет в IT (5 лет в iOS)
- Сверстал 100+ коллекций
- Работал и в больших компаниях
- Сейчас делаю стартап SmileCraft



Почему эта тема интересна

- Мир iOS стоит на ~~котах~~ китах коллекциях
- Не многие знают как с этим работать
- Простыми усилиями можно добиться красивых анимаций



Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Что такое UICollectionView и layout

Существующие подходы работы с layout

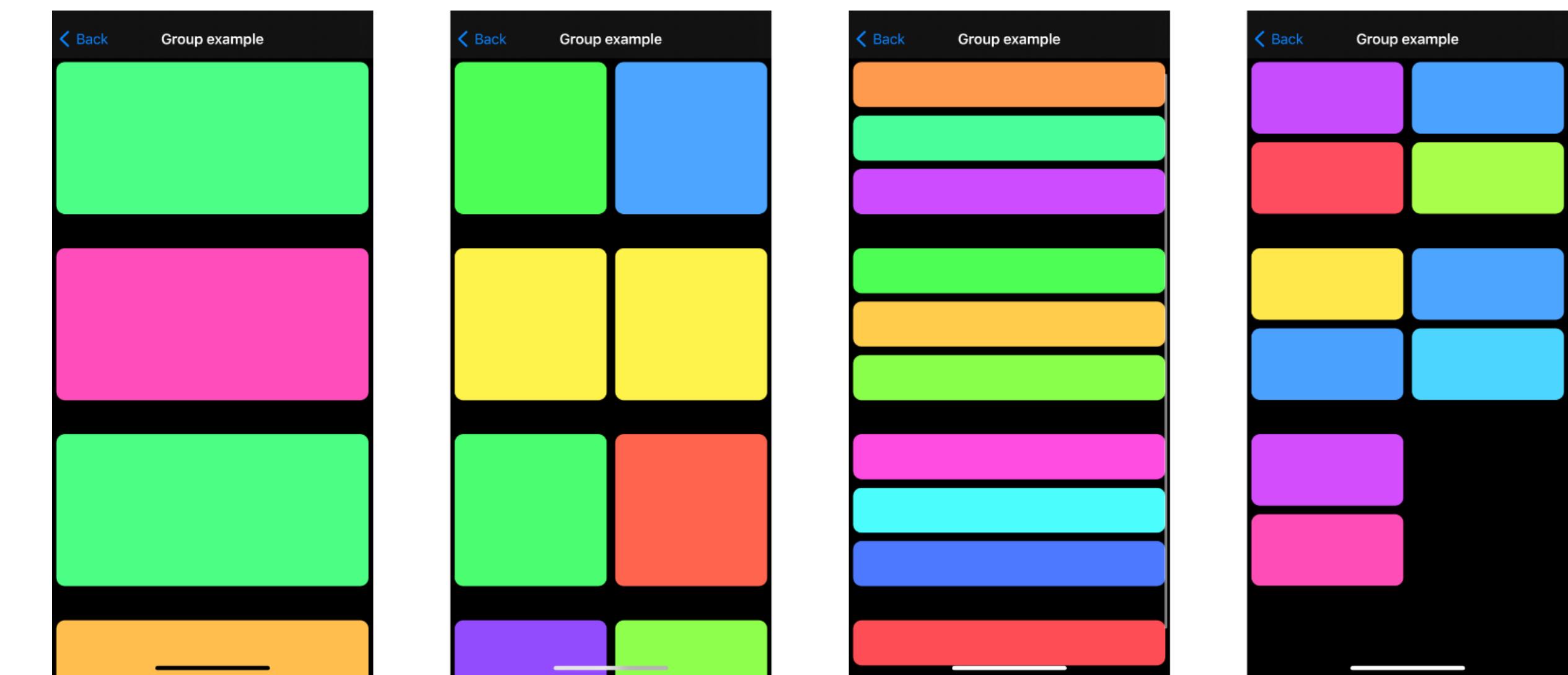
Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Что такое UICollectionView

- Наследник UIScrollView
- Отличается от UITableView тем, что можно задать свой Layout
- Есть reuse ячеек



Что такое UICollectionViewLayout

- Абстрактный класс, от которого мы можем наследоваться
- Отвечает на вопрос «как отображать ячейки и как их скроллить»
- Подготавливает `UICollectionViewLayoutAttributes`, которые коллекция потом просто применяет к ячейкам



Из чего состоит UICollectionViewLayoutAttributes

- frame/bounds/center/size
- transform/transform3D
- alpha
- zIndex
- hidden

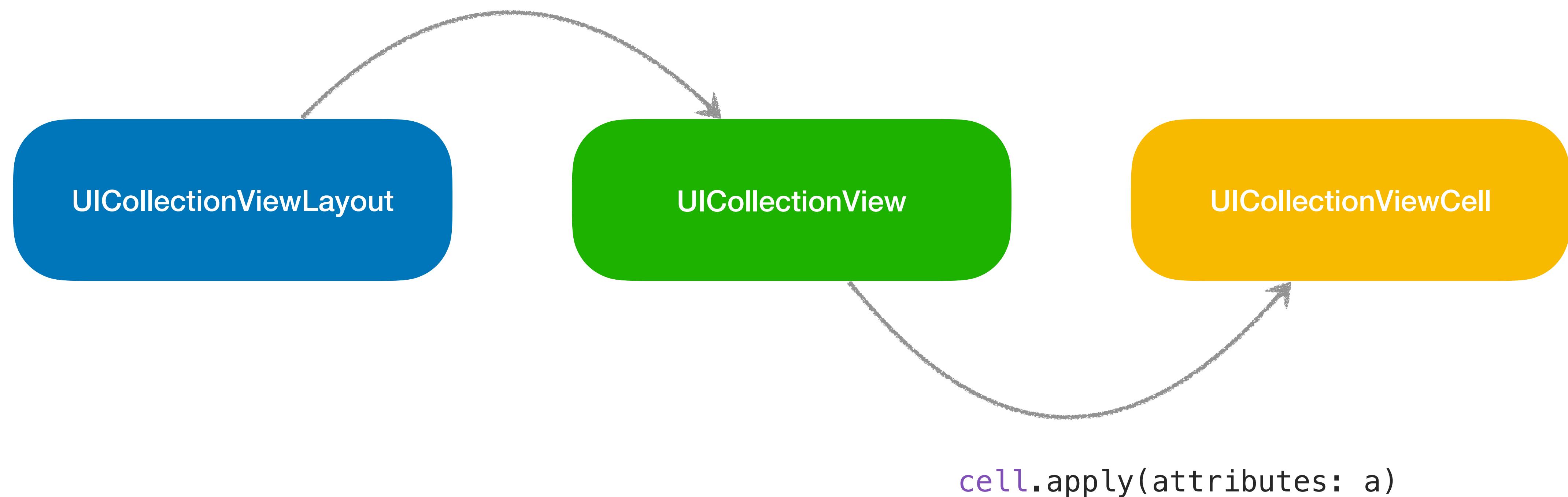
Наша задача при наследовании от UICollectionViewLayout

- Написать логику создания UICollectionViewLayoutAttributes и заполнить все необходимые свойства (frame/zIndex/...)

```
class CustomCollectionViewLayout: UICollectionViewLayout {  
    /* ... */  
  
    override func layoutAttributesForItem(  
        at indexPath: IndexPath  
    ) -> UICollectionViewLayoutAttributes? { /* ... */ }  
  
    override func layoutAttributesForElements(  
        in rect: CGRect  
    ) -> [UICollectionViewLayoutAttributes]? { /* ... */ }  
}
```

Как это все работает вместе

```
let a = layout.layoutAttributes(indexPath: (1, 1))
```



Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Что такое UICollectionView и layout

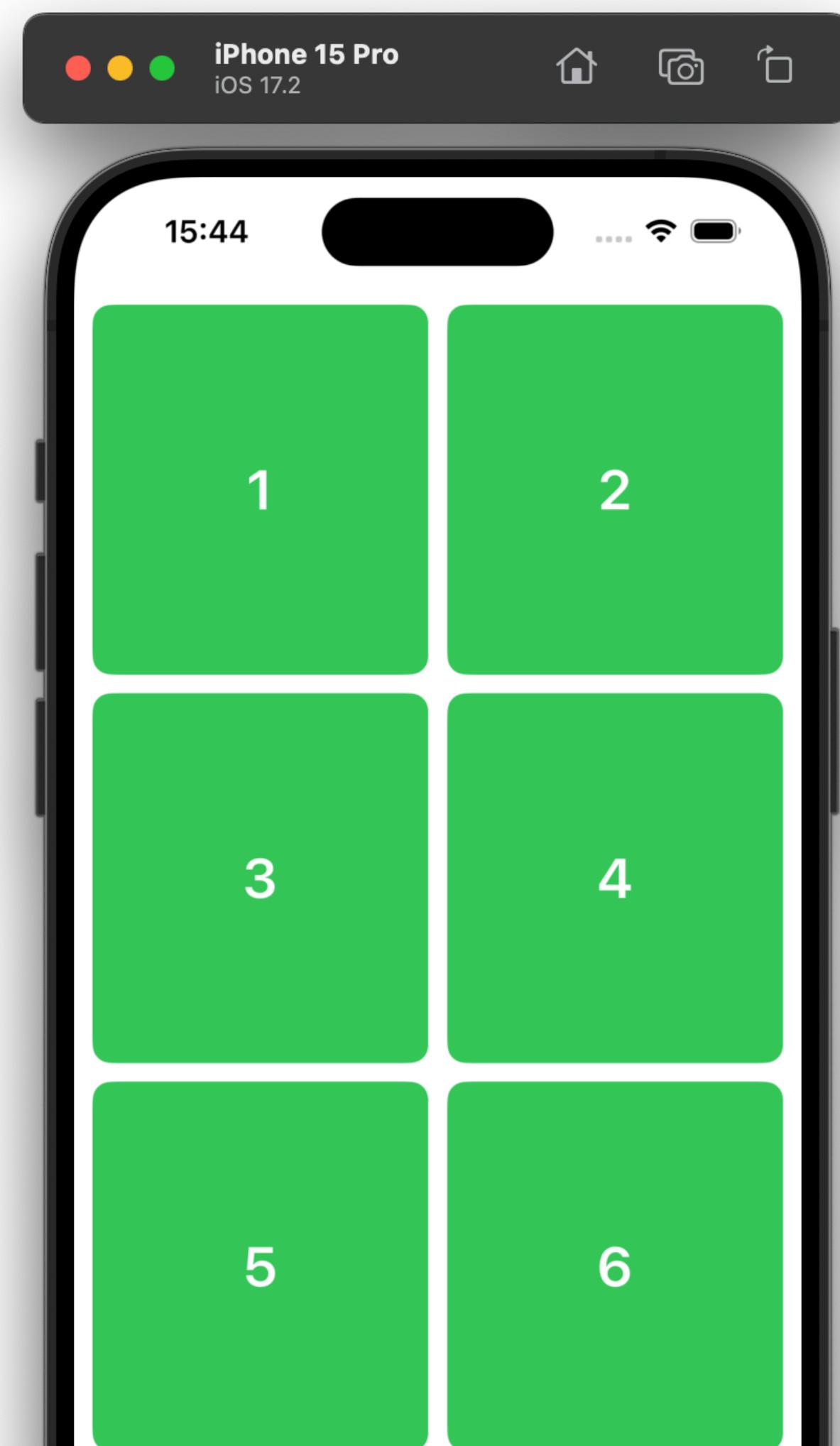
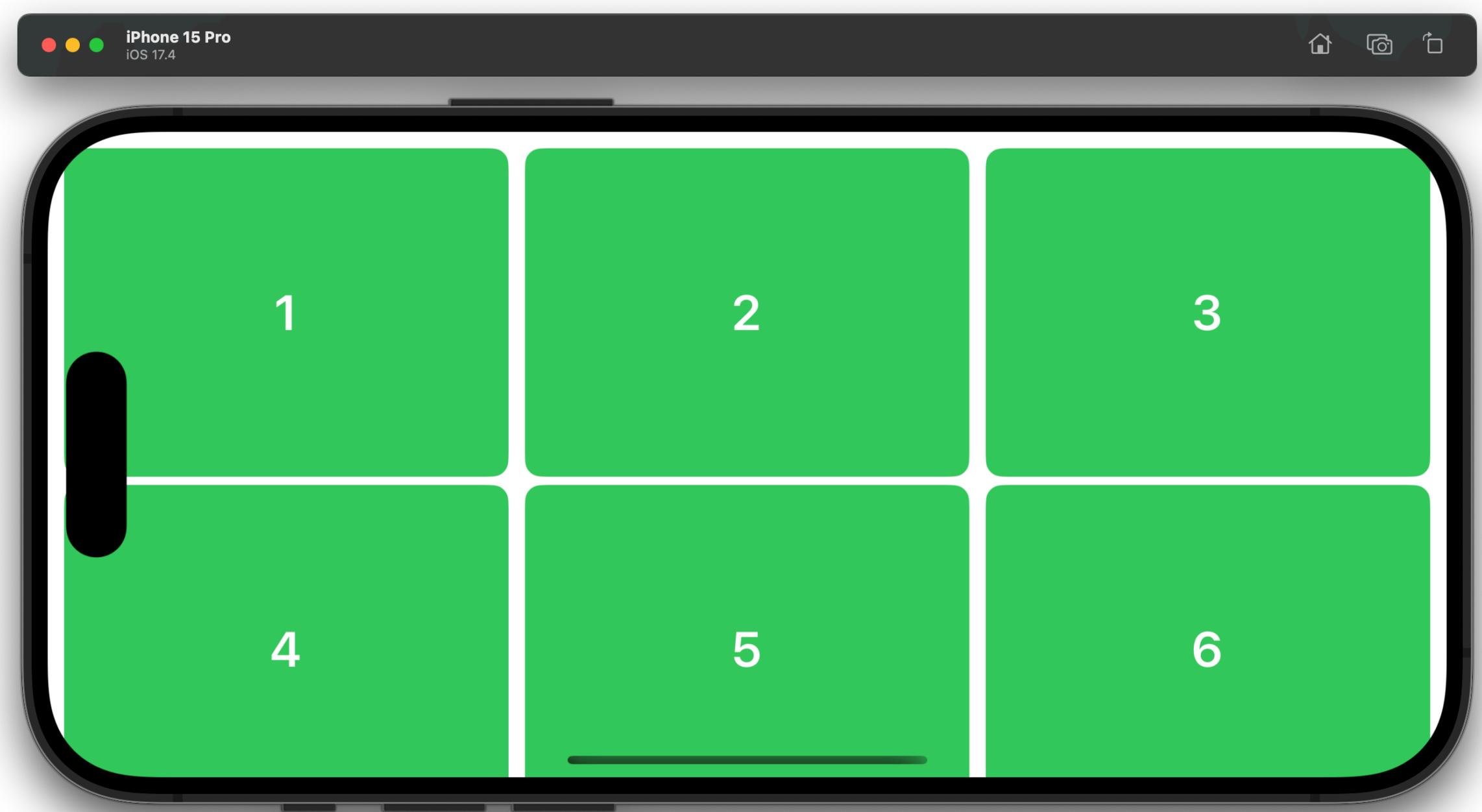
Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

UICollectionViewFlowLayout



UICollectionViewFlowLayout

- Создаем layout и настраиваем

```
let layout = UICollectionViewFlowLayout()
layout.scrollDirection = .vertical
layout.itemSize = .init(width: 100, height: 100)
layout.sectionInset = .init(top: 10, ...)
layout.minimumLineSpacing = 10
layout.minimumInteritemSpacing = 20
```

UICollectionViewFlowLayout advanced

- Создаем layout и настраиваем (базовые вещи)
- Становимся делегатом
- Реализуем делегат UICollectionViewDelegateFlowLayout

```
let layout = UICollectionViewFlowLayout()  
layout.scrollDirection = .vertical
```

UICollectionViewFlowLayout advanced

- Создаем layout и настраиваем
(базовые вещи)
- Становимся делегатом
- Реализуем делегат
UICollectionViewDelegateFlowLayout

```
collectionView.delegate = self
```

UICollectionViewFlowLayout advanced

- Создаем layout и настраиваем (базовые вещи)
- Становимся делегатом
- Реализуем делегат UICollectionViewDelegateFlowLayout

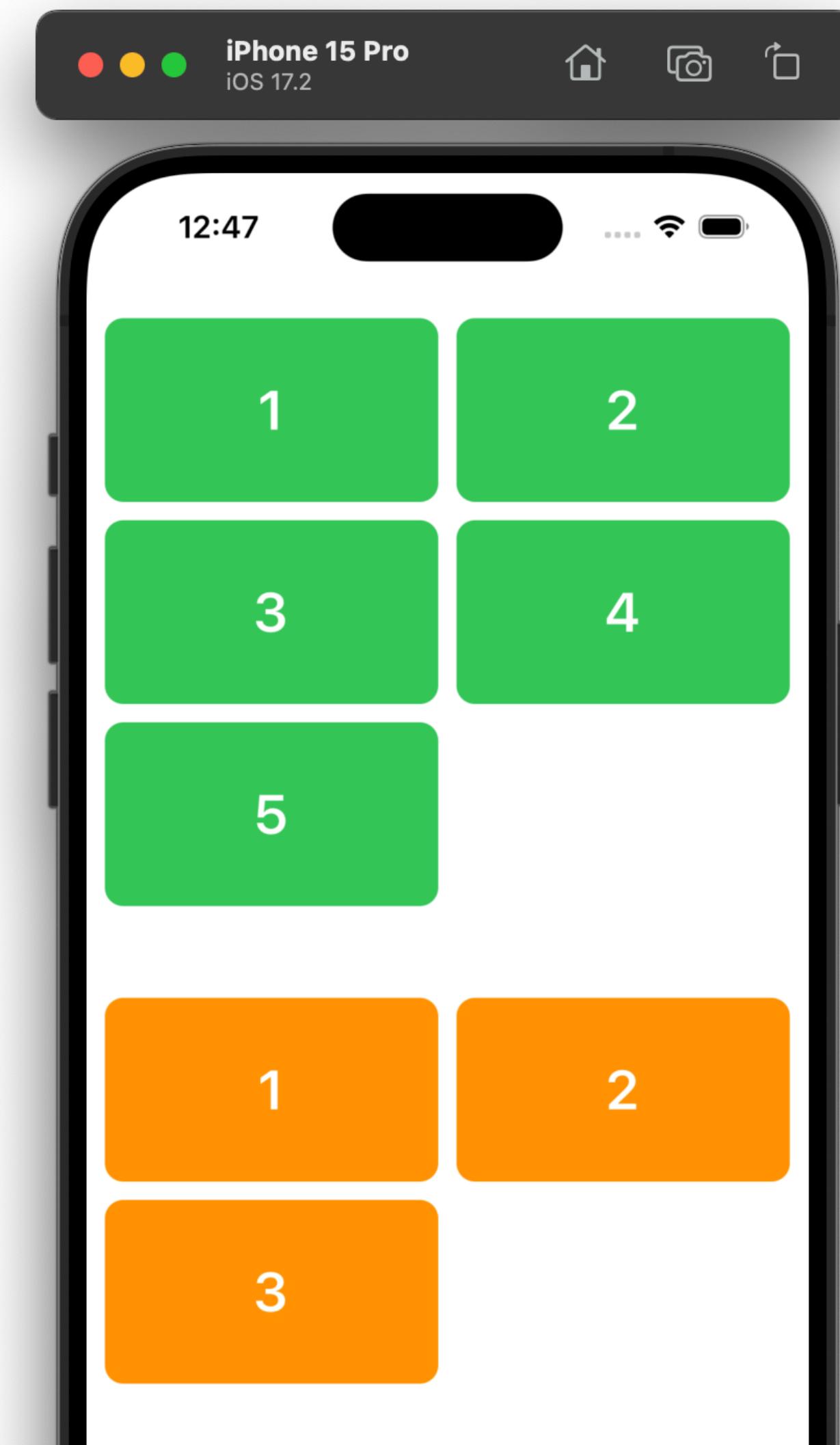
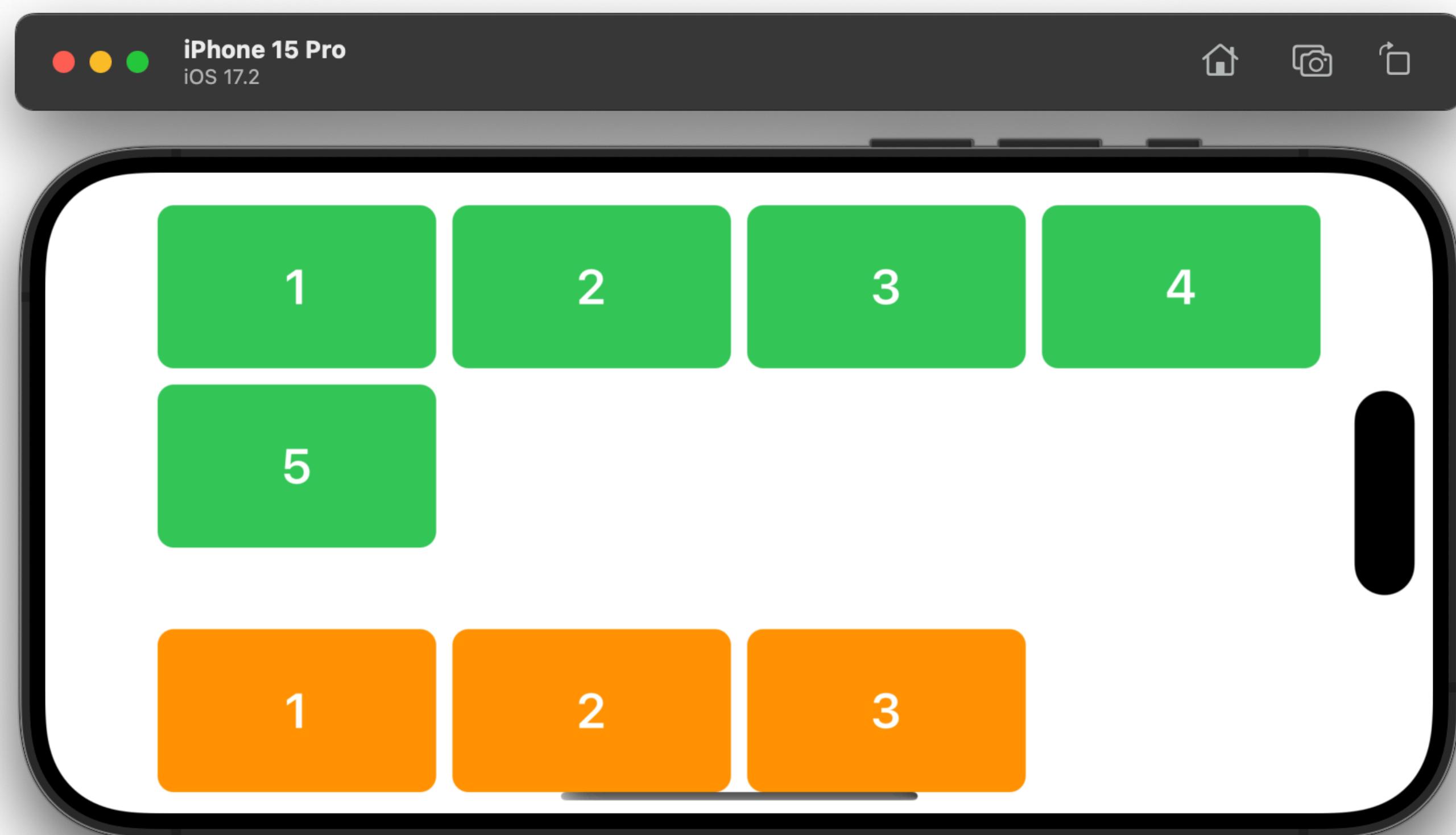
```
func collectionView(  
    collectionView: UICollectionView,  
    layout collectionViewLayout: UICollectionViewLayout,  
    sizeForItemAt indexPath: IndexPath  
) -> CGSize { /* ... */ }  
  
func collectionView(  
    collectionView: UICollectionView,  
    layout collectionViewLayout: UICollectionViewLayout,  
    insetForSectionAt section: Int  
) -> UIEdgeInsets { /* ... */ }  
  
/* ... */
```

UICollectionViewFlowLayout advanced

- Можно задавать отступы и размеры относительно размера коллекции

```
func collectionView(  
    collectionView: UICollectionView,  
    layout collectionViewLayout: ...,  
    sizeForItemAt indexPath: IndexPath  
) -> CGSize {  
    CGSize(  
        width: collectionView.bounds.width,  
        height: collectionView.bounds.height / 2  
    )  
}
```

UICollectionViewCompositionalLayout



UICollectionViewCompositionalLayout

- Создаем layout

```
let layout = UICollectionViewCompositionalLayout(  
    section: section  
)
```

UICollectionViewCompositionalLayout

- Создаем layout
- Создаем секцию

```
let section = NSCollectionLayoutSection(  
    group: mainGroup  
)  
section.contentInsets = .init(  
    top: 20,  
    leading: 10,  
    bottom: 20,  
    trailing: 10  
)
```

UICollectionViewCompositionalLayout

- Создаем layout
- Создаем секцию
- Создаем группу

```
let mainGroup = NSCollectionLayoutGroup.horizontal(  
    layoutSize: .init(  
        widthDimension: .fractionalWidth(1),  
        heightDimension: .absolute(100)  
    ),  
    repeatingSubitem: topItem,  
    count: number0fItemsToRepeat  
)  
mainGroup.interItemSpacing = .fixed(10)  
mainGroup.edgeSpacing = .init(  
    leading: .fixed(0),  
    top: .fixed(0),  
    trailing: .fixed(0),  
    bottom: .fixed(10)  
)
```

UICollectionViewCompositionalLayout

- Создаем layout
- Создаем секцию
- Создаем группу
- Создаем элемент

```
let topItem = NSCollectionLayoutItem(  
    layoutSize: NSCollectionLayoutSize(  
        widthDimension: .fractionalWidth(0.5),  
        heightDimension: .fractionalHeight(1)  
    )  
)
```

UICollectionViewViewCompositionalLayout advanced

- Оборачиваем в новый конструктор, где есть environment

```
let layout = UICollectionViewViewCompositionalLayout { sectionIndex, env in
    /* ... */
    return section
}
```

UICollectionViewCompositionalLayout advanced

- Оборачиваем в новый конструктор, где есть environment
- Добавляем логику расчета количества элементов в строке

```
let width = env.container.effectiveContentSize.width
let num = width > 400 ? 4 : 2

let topItem = NSCollectionLayoutItem(
    layoutSize: NSCollectionLayoutSize(
        widthDimension: .fractionalWidth(1 / CGFloat(num)),
        heightDimension: .fractionalHeight(1)
    )
)
let mainGroup = NSCollectionLayoutGroup.horizontal(
    layoutSize: .init(
        widthDimension: .fractionalWidth(1),
        heightDimension: .absolute(100)
    ),
    repeatingSubitem: topItem,
    count: numberOfItemsToRepeat
)
```

UICollectionViewCompositionalLayout advanced

- Оборачиваем в новый конструктор, где есть environment
- Добавляем логику расчета количества элементов в строке
- Добавляем несколько групп (разберем на Live Coding)

Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Главные методы

- Подготавливаем атрибуты
(кэшируем)

`func prepare()`

Главные методы

- Подготавливаем атрибуты
(кэшируем)
- Возвращаем подготовленные
атрибуты

```
override func layoutAttributesForItem(  
    at indexPath: IndexPath  
) -> UICollectionViewLayoutAttributes? { /* ... */ }  
override func layoutAttributesForElements(  
    in rect: CGRect  
) -> [UICollectionViewLayoutAttributes]? { /* ... */ }
```

Главные методы

- Подготавливаем атрибуты
(кэшируем)
- Возвращаем подготовленные
атрибуты
- Определяем когда будет заново
рассчитываться layout

```
override func shouldInvalidateLayout(  
    forBoundsChange newBounds: CGRect  
) -> Bool { /* ... */ }
```

Главные методы

- Подготавливаем атрибуты
(кэшируем)
- Возвращаем подготовленные
атрибуты
- Определяем когда будет заново
рассчитываться layout
- Возвращаем размер контента

```
override var collectionViewContentSize: CGSize {  
    /* ... */  
}
```

Performance

- Аккуратно смотрим на то, как часто вызывается метод `prepare` (мы это можем контролировать)

`func prepare()`

```
override func shouldInvalidateLayout(  
    forBoundsChange newBounds: CGRect  
) -> Bool { /* ... */ }
```

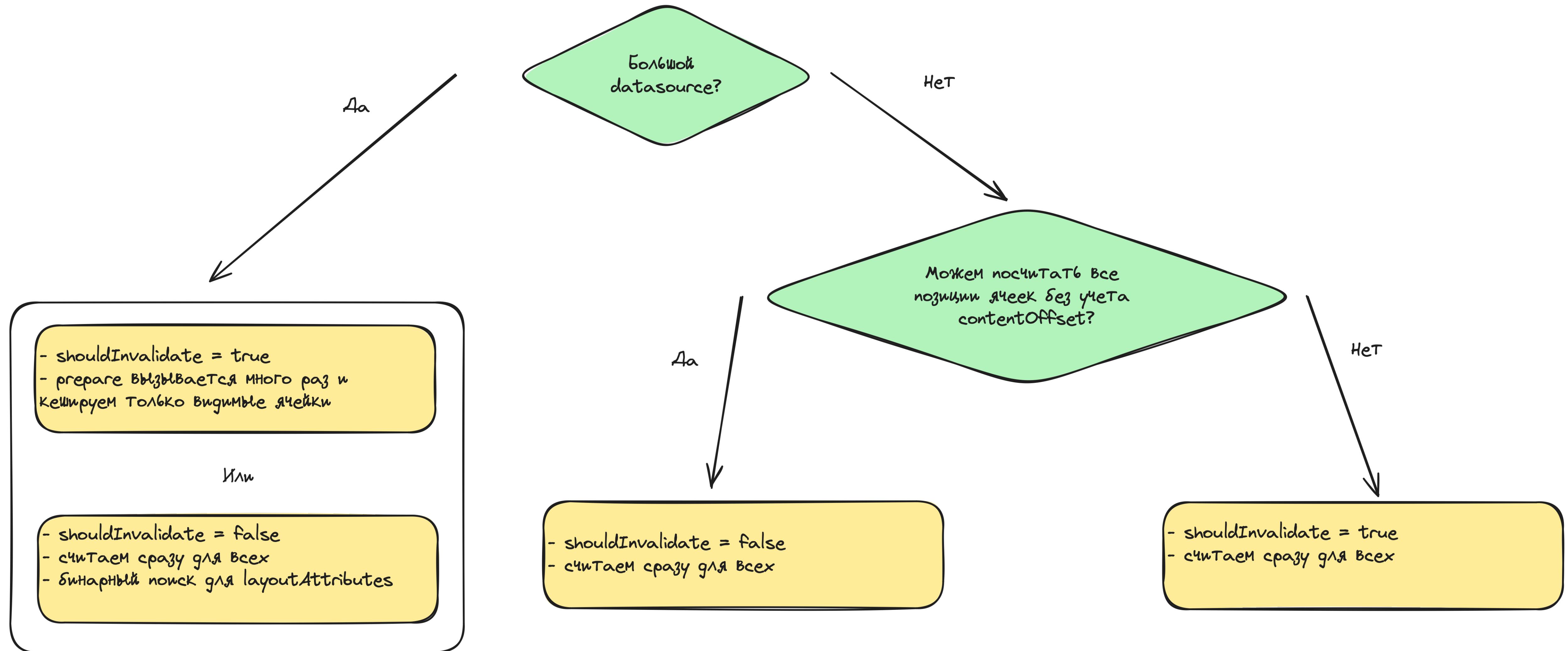
Реfоманс

- Аккуратно смотрим на то, как часто вызывается метод `render` (мы это можем контролировать)
- Если `dataSource` очень большой, то можем высчитывать атрибуты не для всех ячеек
 - В методе `render`, считаем не для всех элементов (на основе отступа коллекции)
 - Инвалидируем `layout` при окончании скролла

Performance

- Аккуратно смотрим на то, как часто вызывается метод `relocate` (мы это можем контролировать)
- Если `dataSource` очень большой, то можем высчитывать атрибуты не для всех ячеек
- Оптимизируем методы (бинарный поиск)

```
override func layoutAttributesForItem(  
    at indexPath: IndexPath  
) -> UICollectionViewLayoutAttributes? { /* ... */ }  
override func layoutAttributesForElements(  
    in rect: CGRect  
) -> [UICollectionViewLayoutAttributes]? { /* ... */ }
```



Infinite Datasource

- Сделать количество элементов очень большим, затем из datasource брать mod от текущего индекса
- Сделать количество элементов не очень большим, но при окончании скролла менять contentOffset

Accessibility

- Не делайте infinite datasource
- Можно отслеживать increment/decrement жест и скроллить вручную

Page Scroll

- Переопределить метод

```
override func targetContentOffset(  
    forProposedContentOffset proposedContentOffset: CGPoint,  
    withScrollingVelocity velocity: CGPoint  
) -> CGPoint
```

Animations

- UIKit dynamics
- Attributes

`func initialLayoutAttributesForAppearingItem`

`func finalLayoutAttributesForDisappearingItem`

Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Когда выбрать Compositional

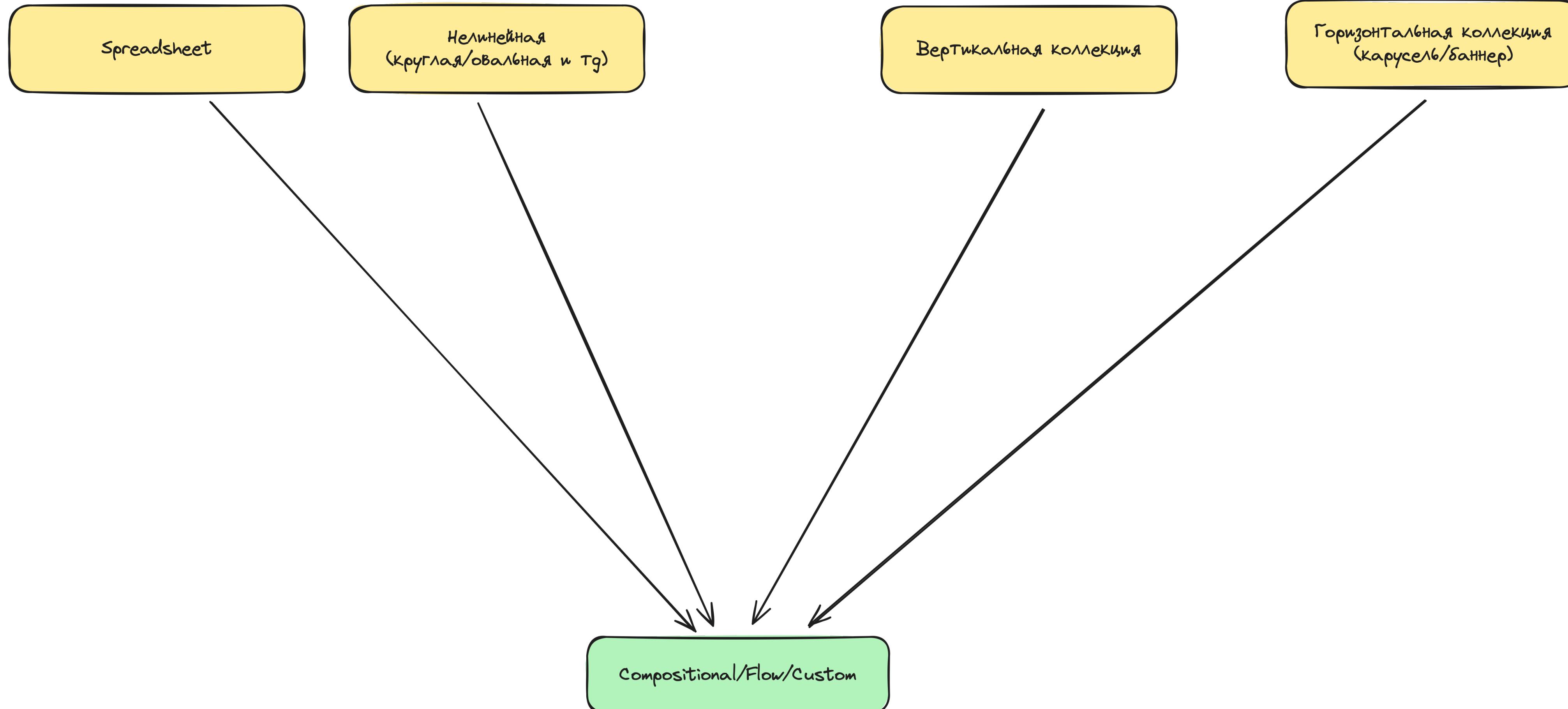
- Хорошо
 - Повторяющиеся паттерн
 - Вертикальная коллекция с горизонтальными секциями
- Плохо
 - Анимации горизонтального скролла
 - Сложные неповторяющиеся паттерны
 - Нелинейный layout

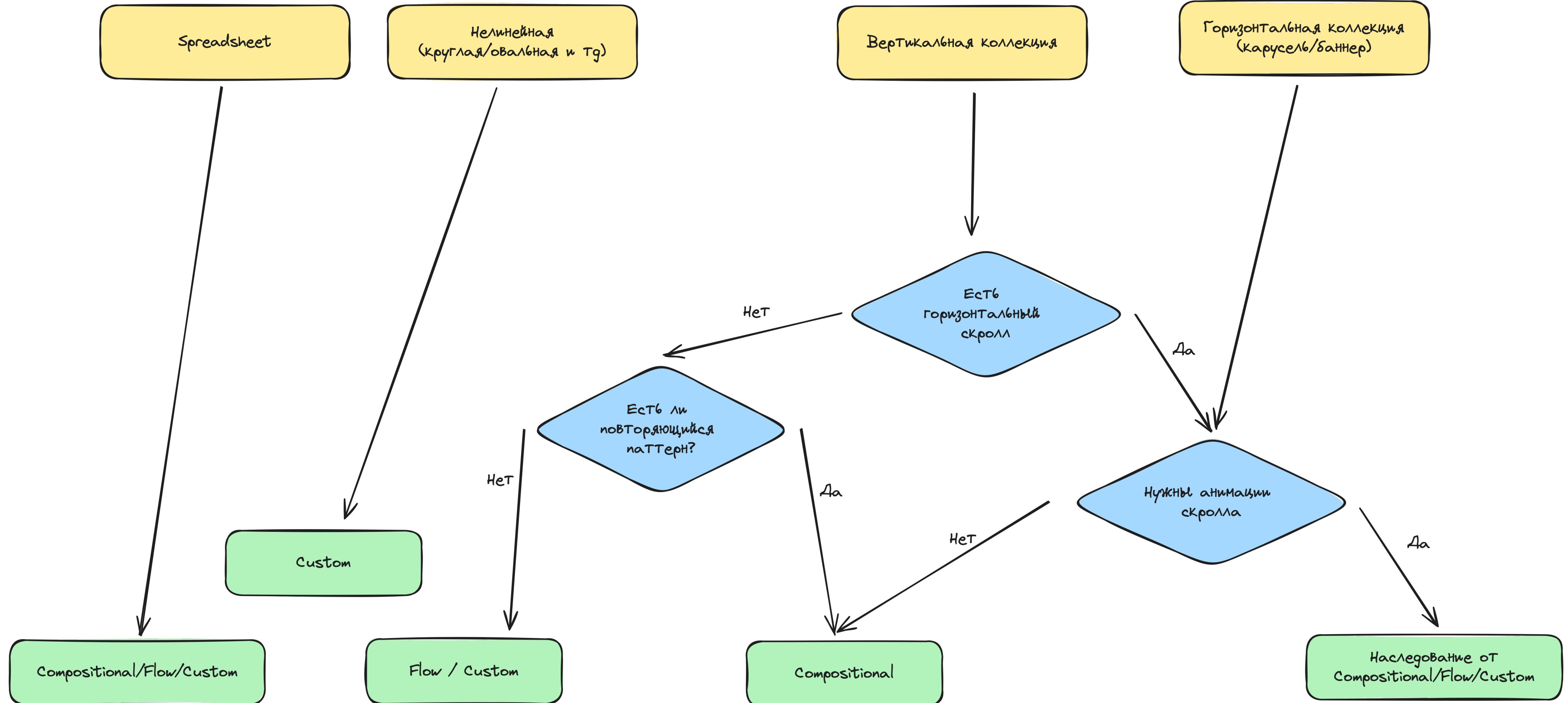
Когда выбрать Flow

- Хорошо
 - Простой случай (вертикальный/горизонтальный)
 - Легко переопределять и подтюнить attributes под себя
- Плохо
 - Размер ячейки по размеру коллекции

Когда выбрать Custom

- Хорошо
 - Полный контроль
- Плохо
 - Полный контроль





Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

Что такое UICollectionView и layout

Существующие подходы работы с layout

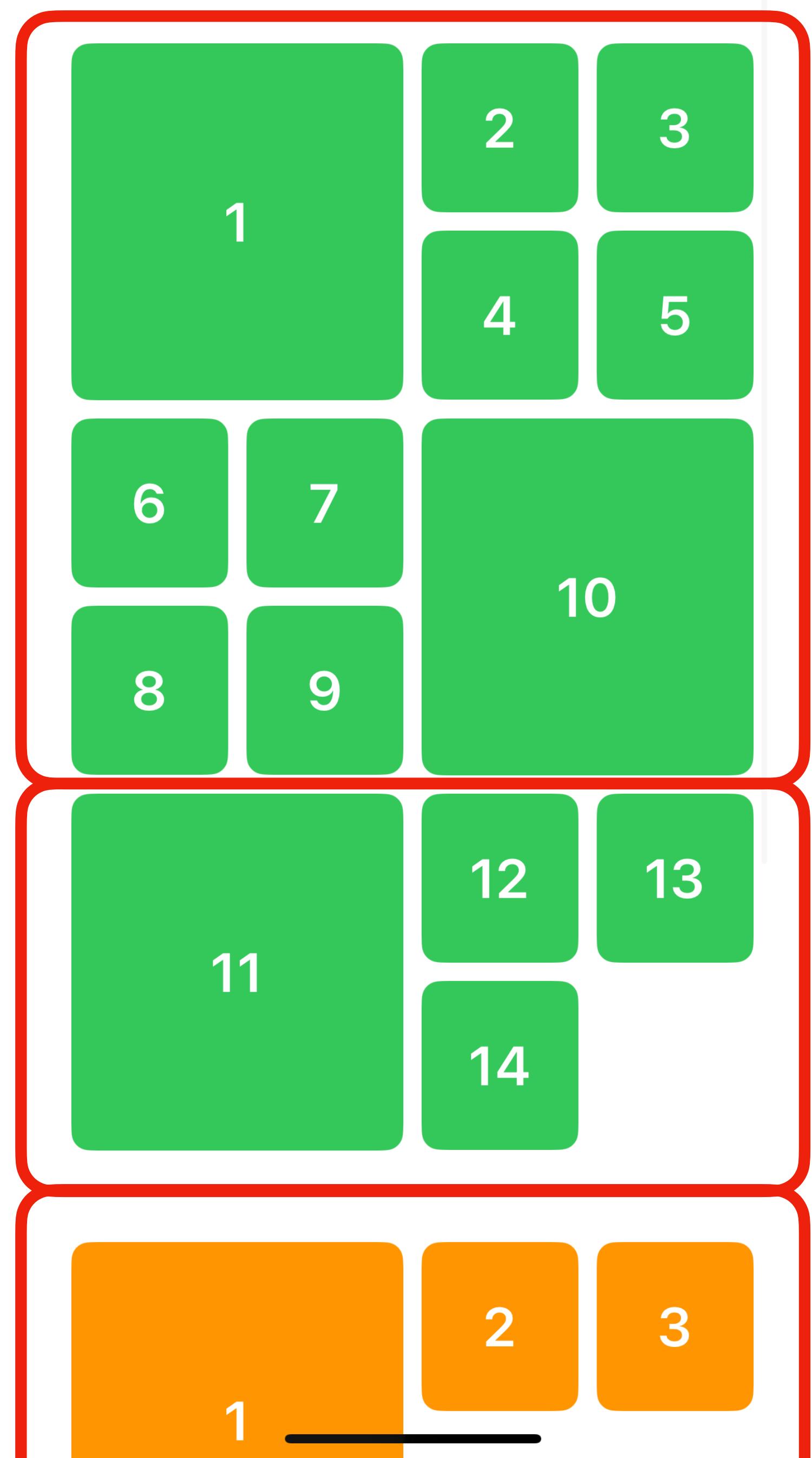
Учимся готовить UICollectionViewLayout

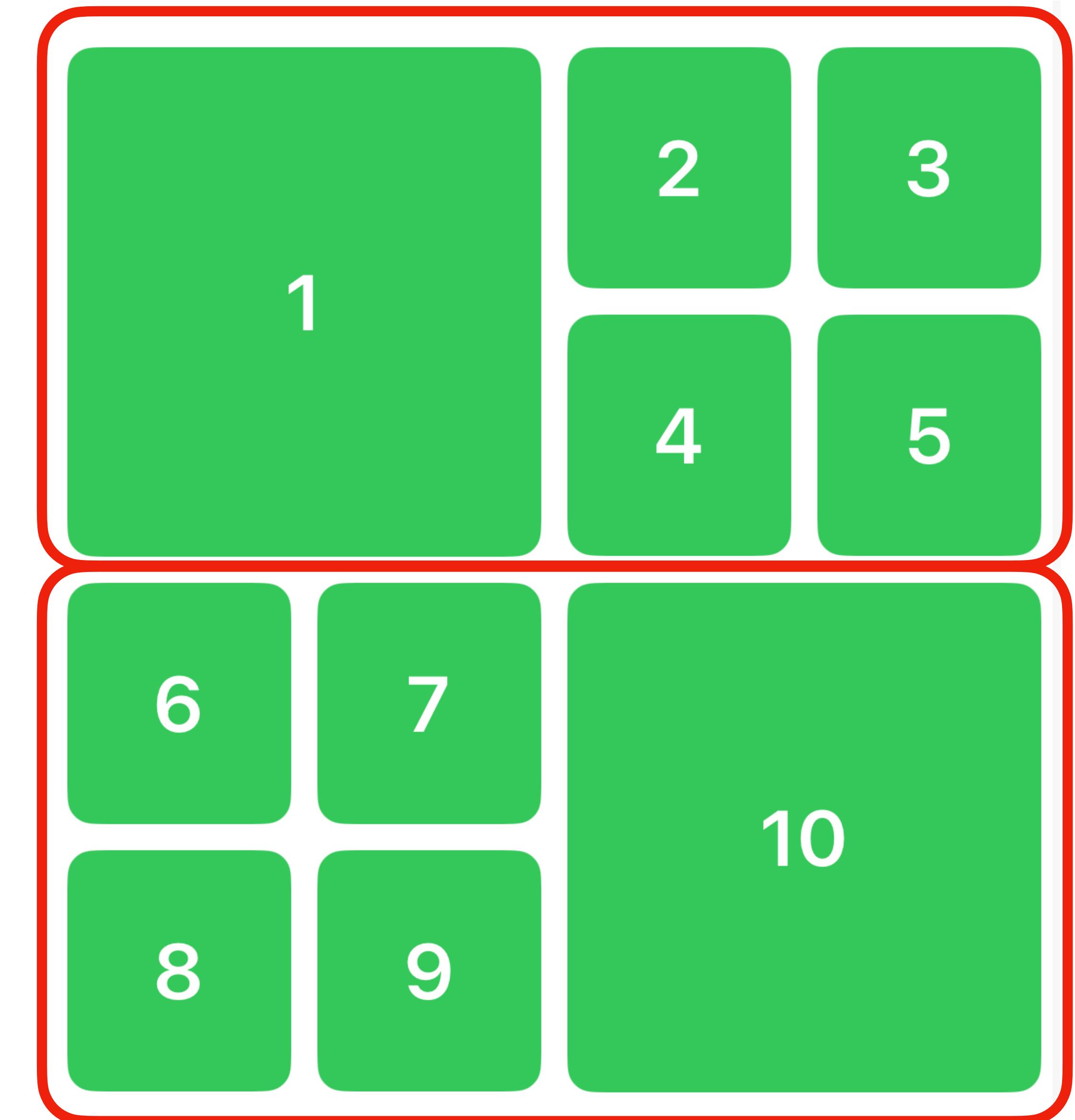
Когда какой layout выбрать

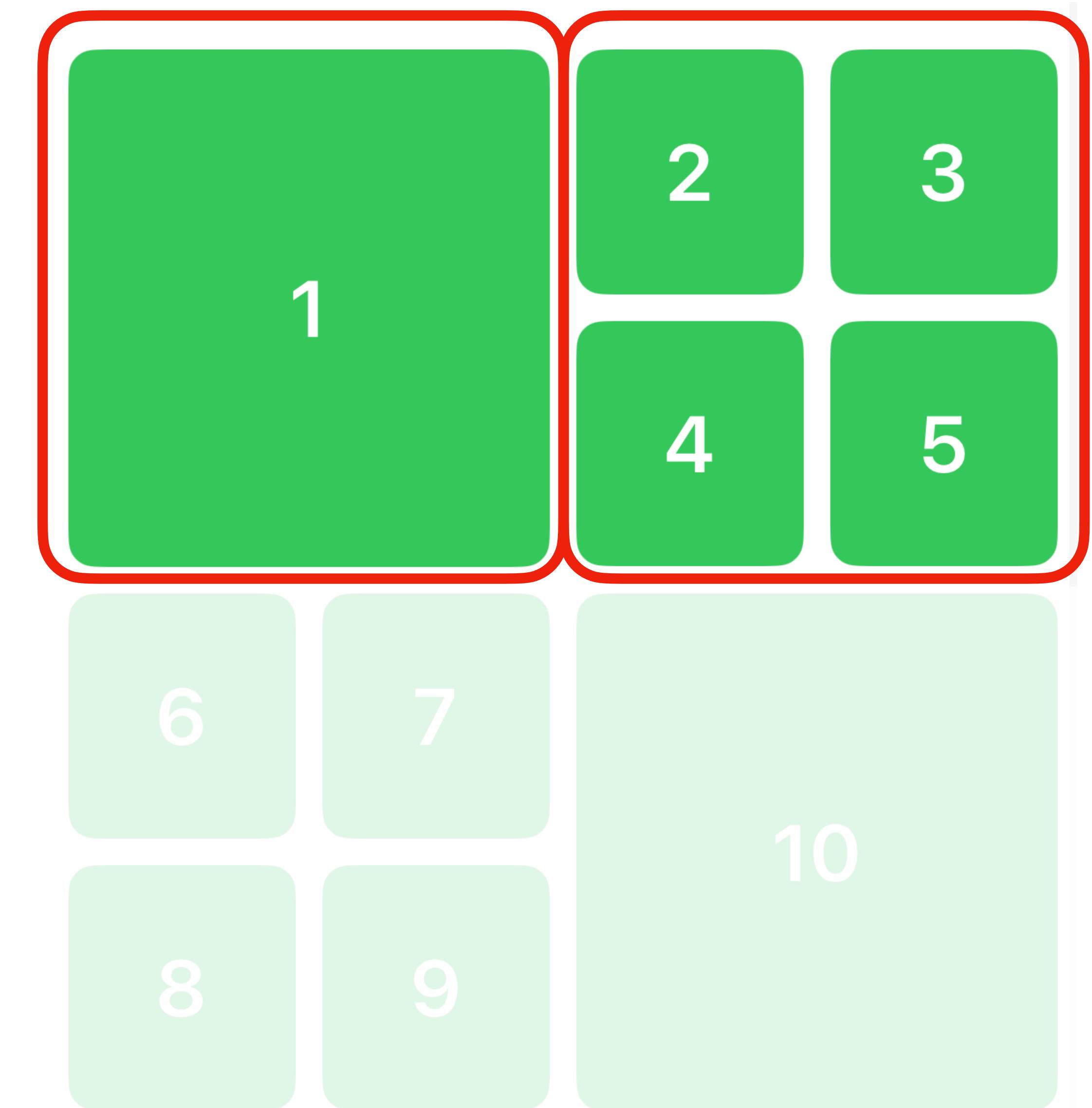
Live coding

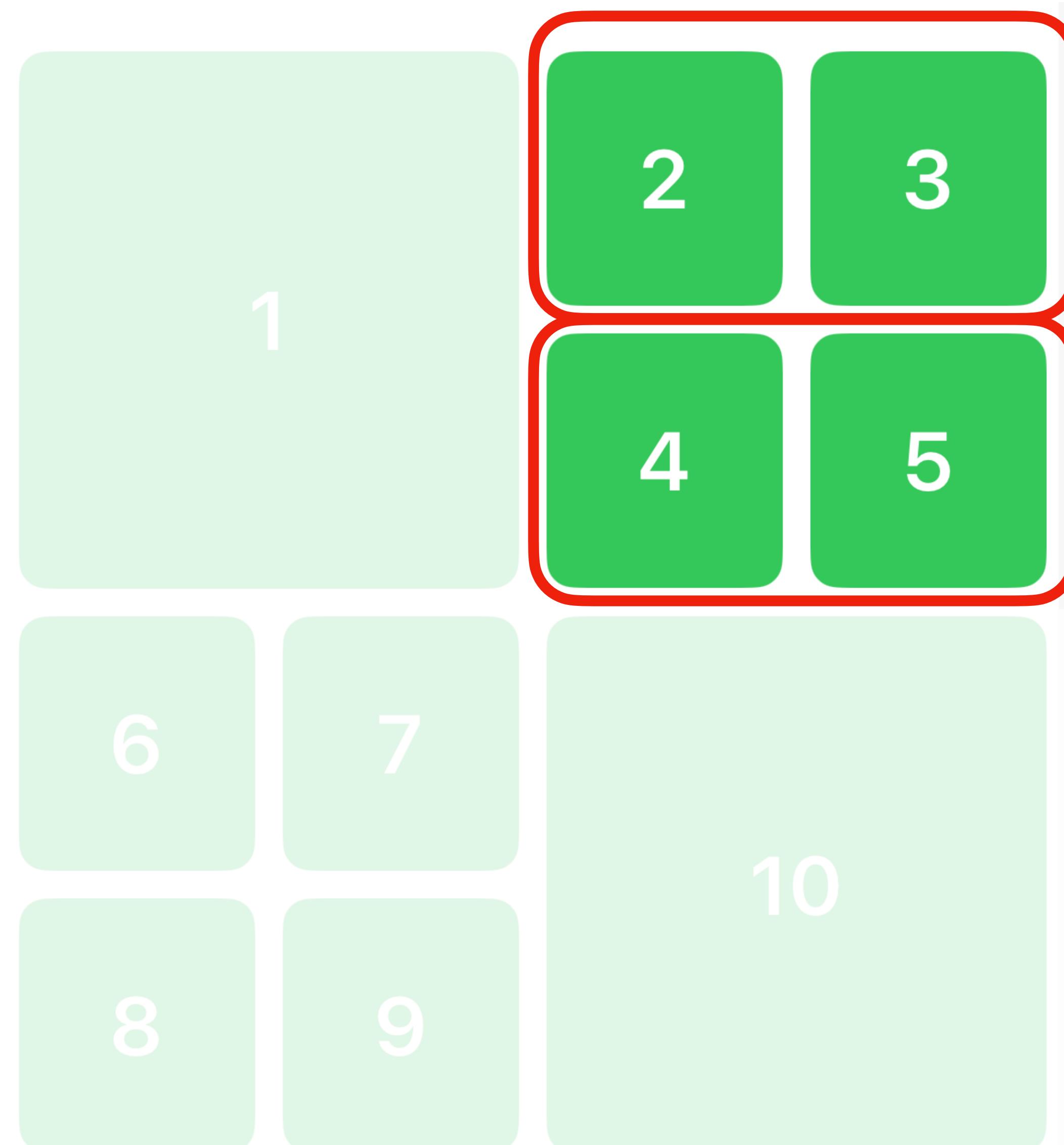
CompositionalLayout

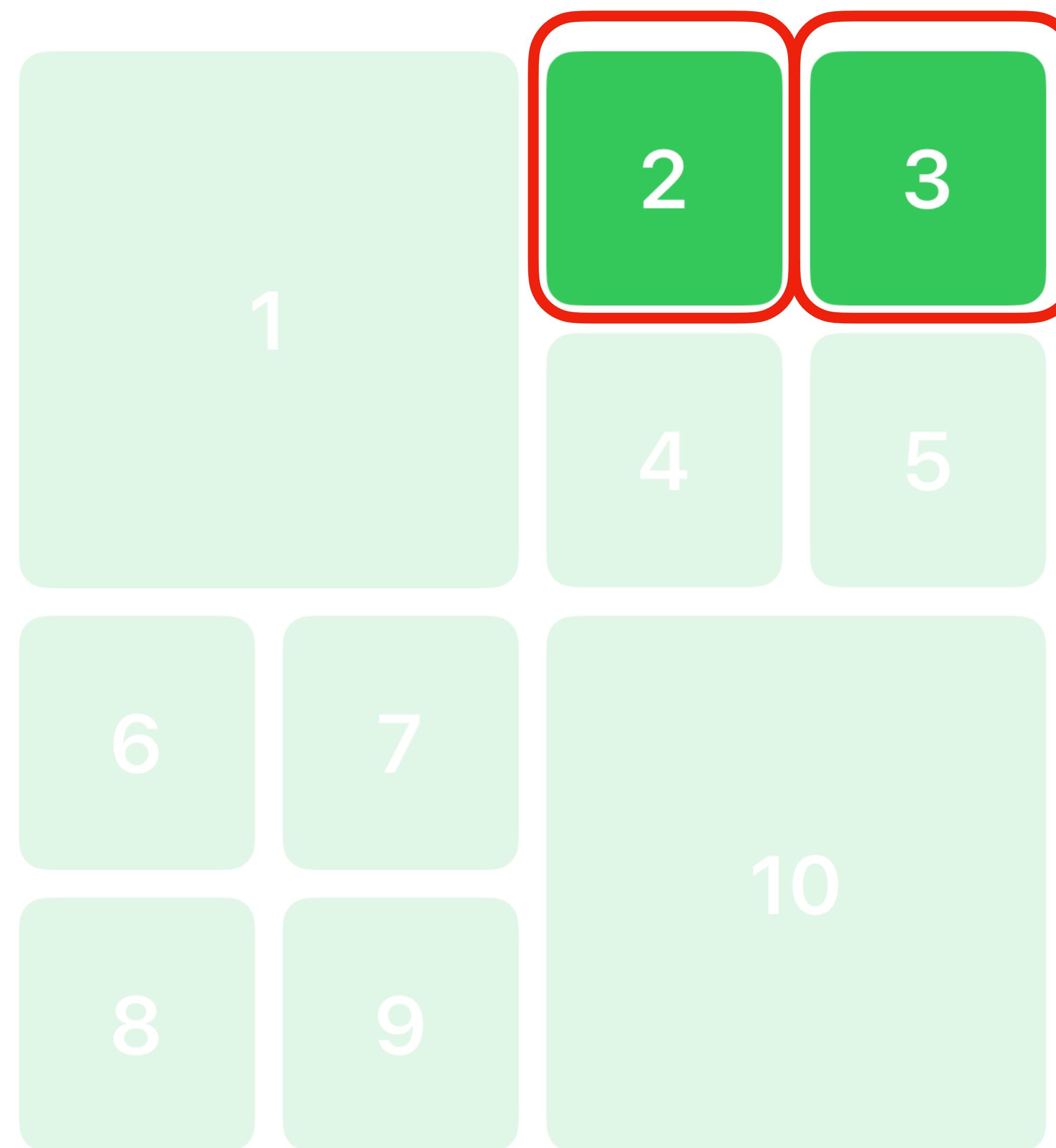








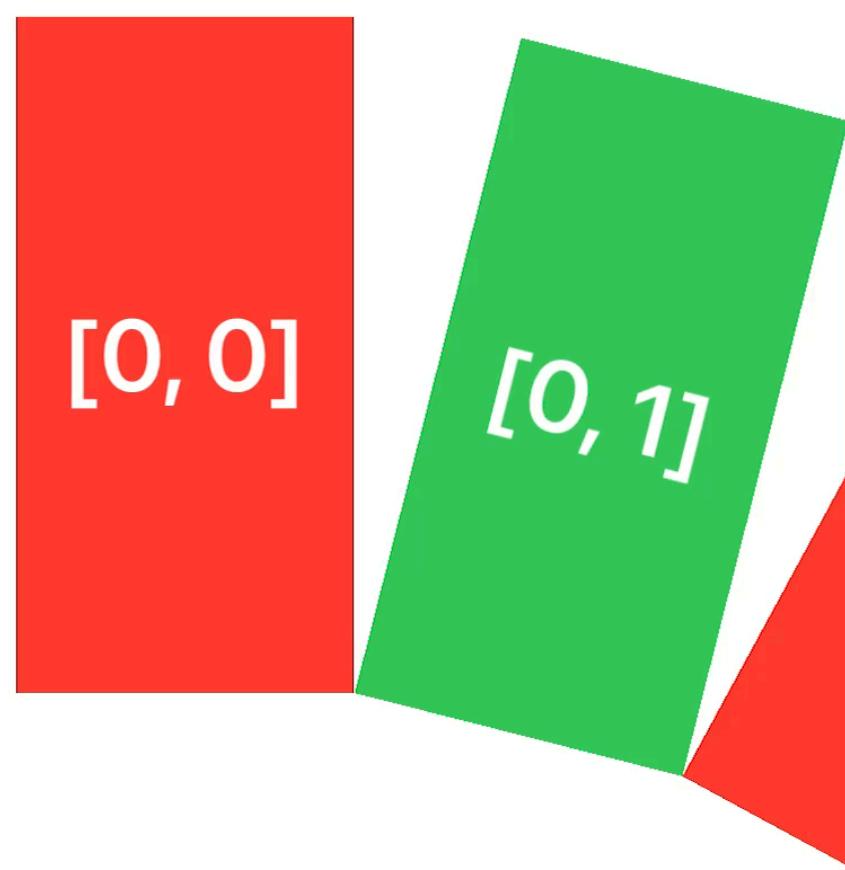


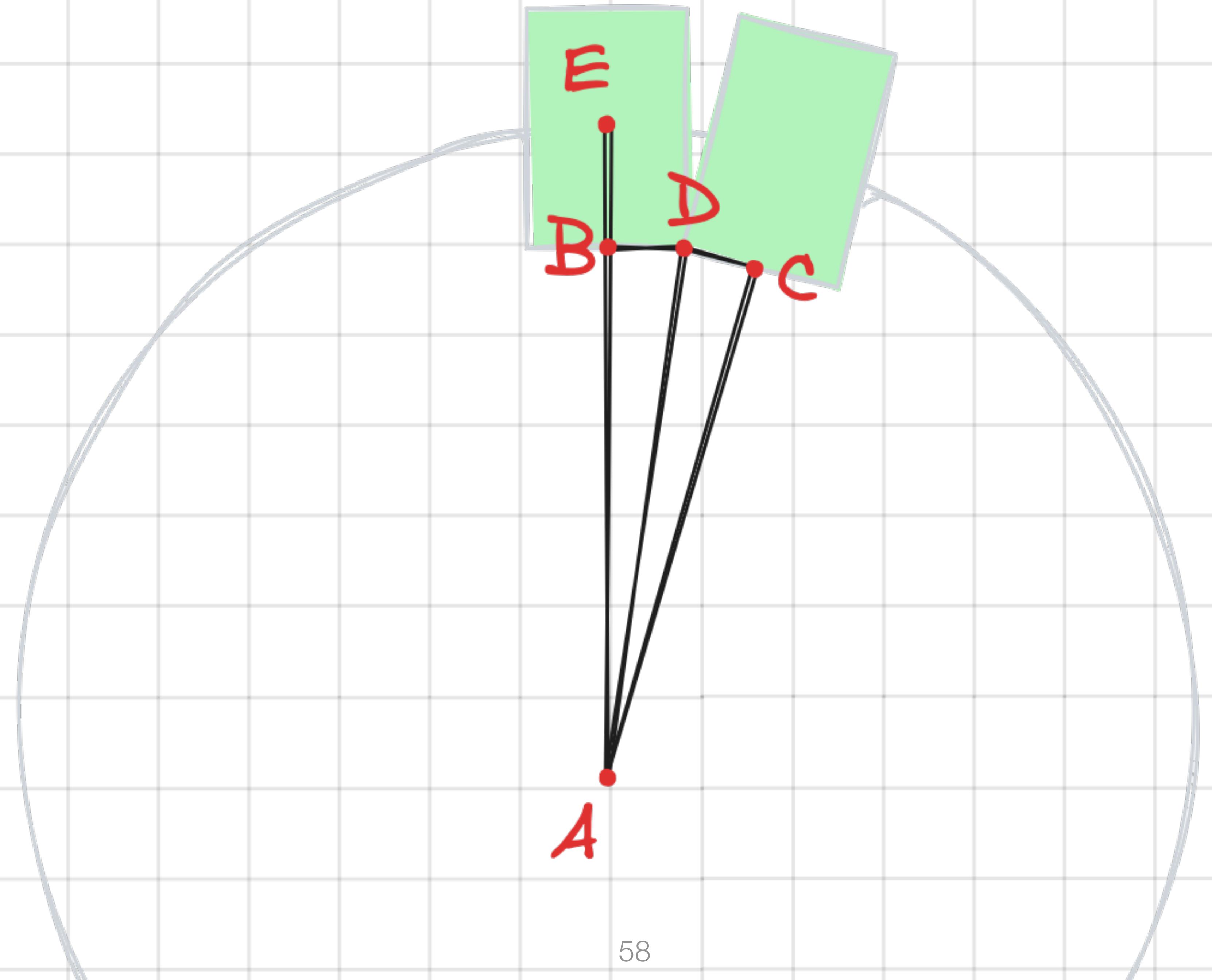


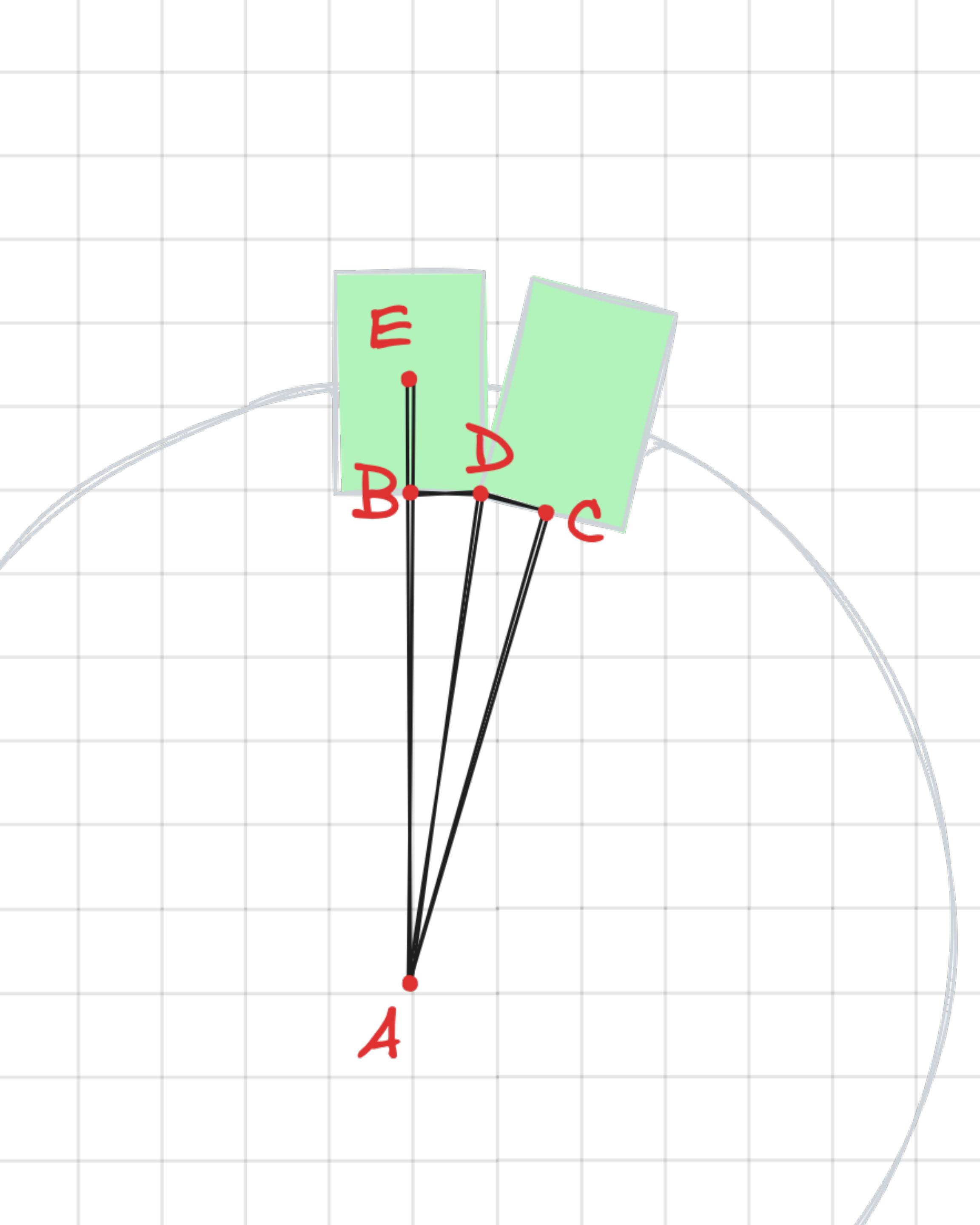
FlowLayout

[0, 5]

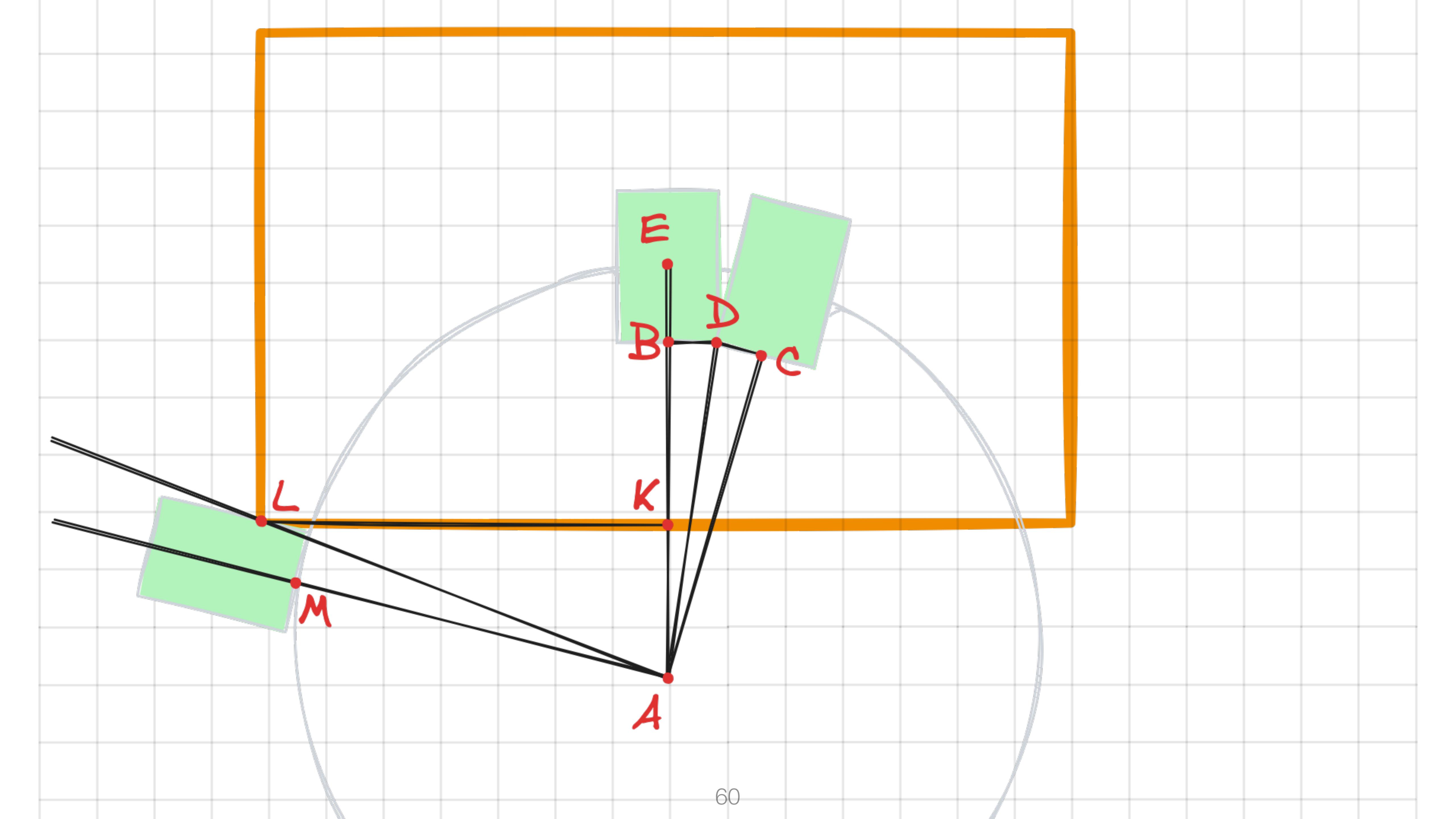
Wheel Layout

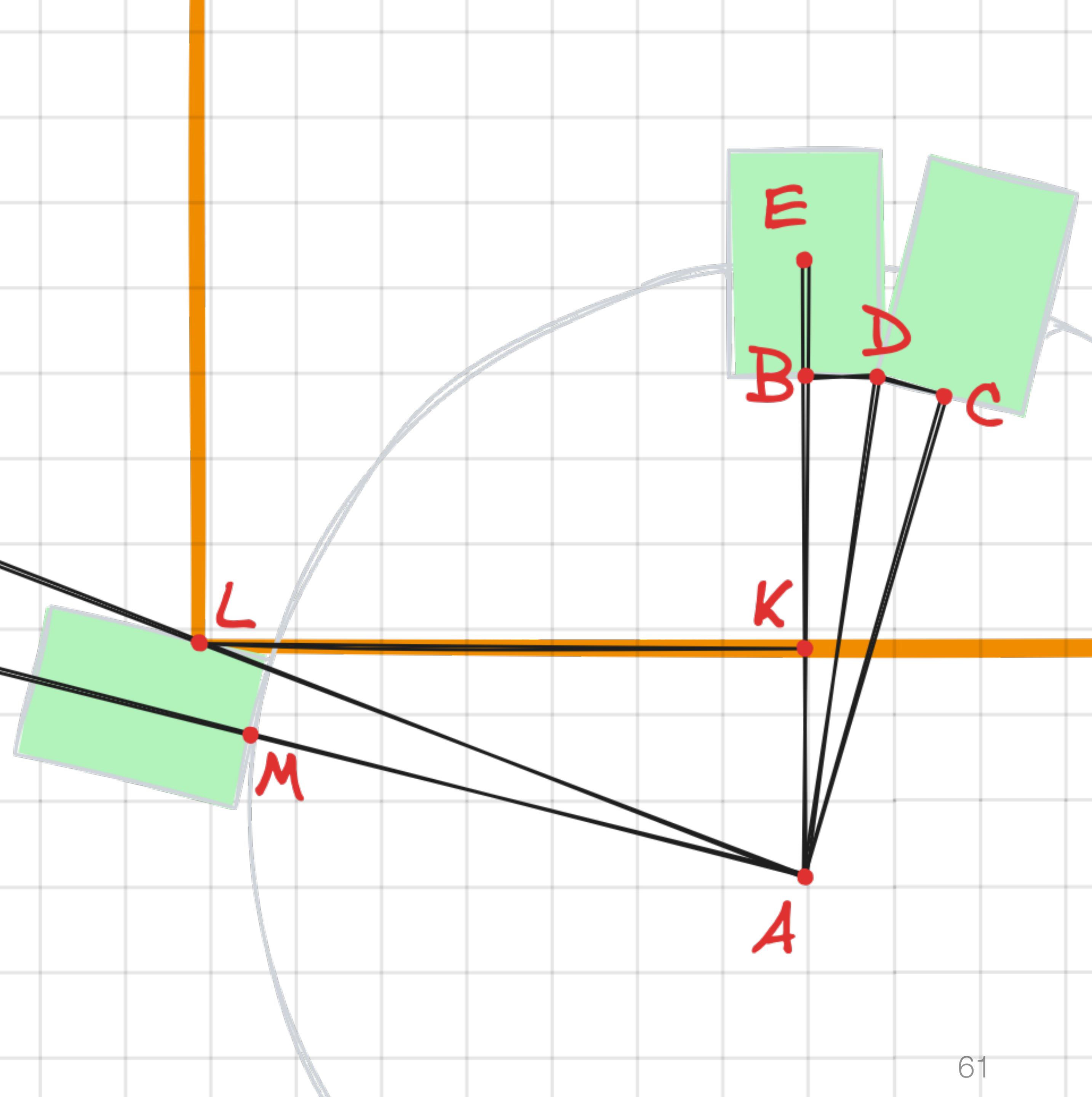






$$\angle BAD = \arctan \frac{BD}{AB} = \arctan \frac{\text{itemSize.width}/2}{r - \text{itemSize.height}/2}$$





$$\begin{aligned}\angle MAE &= \angle MAL + \angle LAE = \\&= \text{itemAngle} + \arctan \frac{LK}{KA} = \\&= \text{itemAngle} + \arctan \frac{\text{screenW}/2}{r - \text{screenH}/2}\end{aligned}$$

Что такое UICollectionView и layout

Существующие подходы работы с layout

Учимся готовить UICollectionViewLayout

Когда какой layout выбрать

Live coding

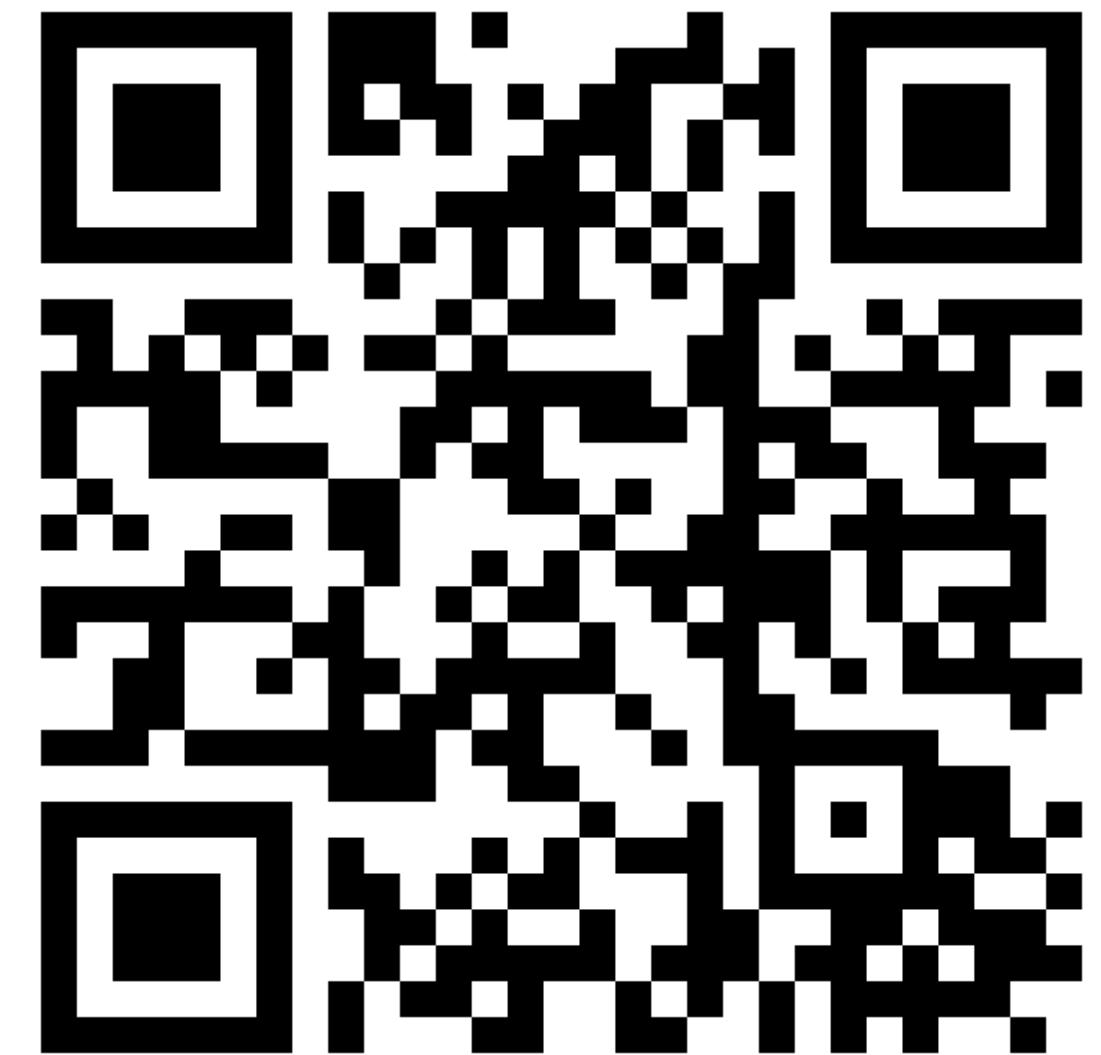
Источники

- WWDC
- Github репозитории
- Код из этого воркшопа
- Статьи



Александр Филимонов

@alexfilimon



UICollectionViewLayout: от каменного века до Compositional Layout