





# Animaciones y Física realistas con UIKit Dynamics

Rodrigo Nicolás Mato



# Agenda

- Definición
- Arquitectura
- Behaviours Predefinidos
- Behaviours Custom
- Conclusiones



- Motor completo de física
- Animaciones + UIKit
- Orientado a la funcionalidad
- Realismo + simpleza



- Opciones previas:
  - Core Animation
  - [UIView animateWithDuration:...]
  - [UIView animateKeyframesWithDuration:...]
  - Motion Effects
  - Sprite Kit



# Agenda

- Definiciones ✓
- Arquitectura
- Behaviours Predefinidos
- Behaviours Custom
- Conclusiones



Arquitectura



- UIDynamicAnimator
  - Control del motor de física
  - Contexto + Sistema de coordenadas. ( Reference View )
  - Producir las animaciones.

```
animator = [[UIDynamicAnimator alloc] initWithReferenceView:referenceView];  
  
[animator addBehavior:...];  
[animator addBehavior:...];
```



- UIDynamicBehaviour
  - Comportamiento físico
  - 7 predefinidos + Custom Behaviours
  - Asociado a 1, o más, items + 1 animator
  - Agregar/remover dinamicamente

```
myBehavior = [[MyBehavior alloc] initWith...];  
[animator addBehavior:myBehavior];
```



- UIDynamicItem
  - Protocolo (UIView lo implementa)
  - Para asociar objetos a Behaviours
  - Acceso a bounds - center - transform

```
@protocol UIDynamicItem <NSObject>
```

```
@property (nonatomic, readwrite) CGPoint center;
```

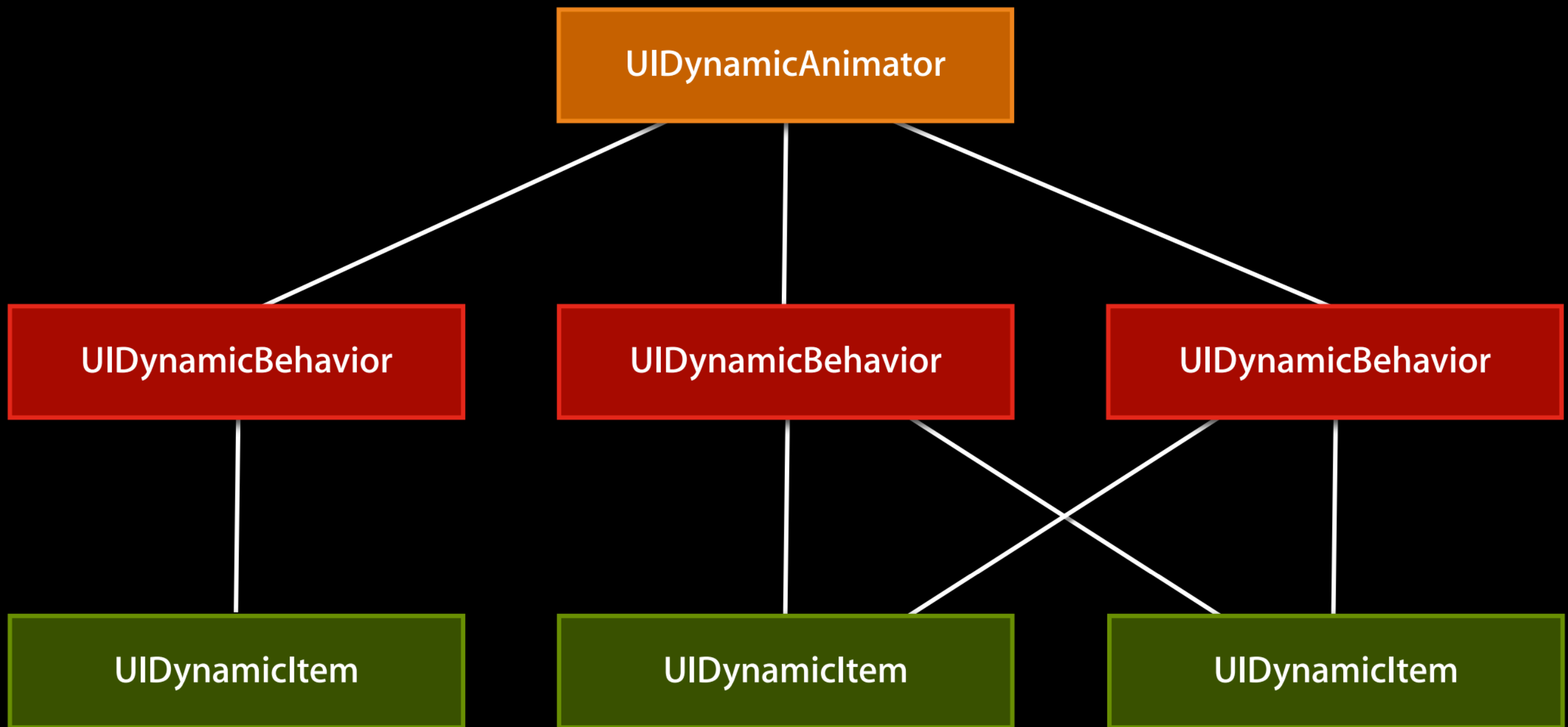
```
@property (nonatomic, readonly) CGRect bounds;
```

```
@property (nonatomic, readwrite) CGAffineTransform transform;
```

```
@end
```



# Arquitectura





# Agenda

- Definiciones ✓
- Arquitectura ✓
- Behaviours Predefinidos
- Behaviours Custom
- Conclusiones



# Behaviours Predefinidos

- UIGravityBehavior
- UICollisionBehavior
- UIPushBehavior
- UIAttachmentBehavior
- UISnapBehavior
- UIFieldBehaviour
- UIDynamicItemBehavior



*DEMO*



# Agenda

- Definiciones ✓
- Arquitectura ✓
- Behaviours Predefinidos ✓
- Behaviours Custom
- Conclusiones



# Behaviours Custom

- Encapsular varios behaviours en uno
- Subclassing - combinables - agrupamiento
- Acciones en cada paso de las animaciones
- Contemplar posibles conflictos



- Ejemplos:

- Bounce -> Gravity + Collision
- Lock Screen -> Gravity + Collision + Attachment + Push
- Drag & Snap -> Attachment + Snap
- Magnet -> Push + Push



*DEMO*



# Agenda

- Definiciones ✓
- Arquitectura ✓
- Behaviours Predefinidos ✓
- Behaviours Custom ✓
- Conclusiones



# Conclusiones

- Focalizar el desarrollo en la User Experience
- Aprovechar la simpleza de implementación para agregar Dynamics a las interacciones
- Componer y extender los behaviours predefinidos para desarrollar efectos é interacciones de alta complejidad
- Encapsular y definir nuestra propia API de Behaviours
- Tener en cuenta conflictos entre propiedades e Impactos de performance



Preguntas?



Muchas Gracias.