



Desenvolvimento para iPhone

Usando Objective-C e iOS
SDK 8

Agenda

- Comentários sobre a WWDC'15;
- Animação avançada:
 - UIDynamicAnimator;
 - UIDynamicBehavior;
- Hora de Brincar!

Objetivos do Dia

- Compreender as diferenças entre a animação da aula passada e a animação da aula de hoje;
- Conhecer os meios para aplicar o UIDynamicAnimator.

Comentários sobre a WWDC'15



WWDC'15

- Novidades do iOS 9;
- Swift 2.0 e o futuro;
- App Thinning;
- Novidades no Objective-C;
- watchOS 2 e apps nativos;
- iPhone 4S receberá o iOS 9 (quebra de paradigma);
- Novo Apple Developer Program integrado!
- Xcode 7
 - Não precisará mais da conta paga para testar no próprio device!

WWDC'15

- Resumo:
- <https://www.youtube.com/watch?v=YRzPhppLYeM>
- Geral:
- <https://developer.apple.com/videos/wwdc/2015/>
- Telas comentadas do iOS 9
- <https://macmagazine.com.br/2015/06/09/confira-o-nosso-primeiro-compilado-de-screenshots-com-novidades-ios-9/>

Animação Avançada (UIDynamicAnimator)

- Abordagem diferente em relação ao que vimos semana passada;
- Como?
 - Configura-se “atributos” relacionados à física dos objetos;
 - Em seguida, permite-se que os objetos animem (de acordo com a “física” definida) e, por fim, cheguem a um estado de equilíbrio;
 - É possível evitar o estado de equilíbrio, mas pode levar a problemas de performance;

UIDynamicAnimator

- Passo-a-passo:
 1. Criar um objeto do tipo UIDynamicAnimator;
 2. Adicionar a ele UIDynamicBehaviors (gravidade, colisões, etc);
 3. Adicionar UIDynamicItems (UIView, normalmente) ao UIDynamicBehavior;
 4. Plim! Magia acontece!

UIDynamicBehavior

- **UIGravityBehavior:**
 - Ângulo;
 - Magnitude ($1.0 = 1000 \text{ pt/s}^2$);
- **UICollisionBehavior**
 - Modo;
 - Boundary (“em quais lados” ocorre);
- **UIAttachmentBehavior**
 - A um item;
 - A uma âncora;
 - Ao centro;

UIDynamicBehavior

- **UISnapBehavior**

- Imagine quatro molas, uma em cada lado da tela. Agora você tem o controle individual sobre o “amortecimento” de cada uma; (damping)

- **UIPushBehavior**

- Modo (contínuo ou instantâneo);
- Direção;
- Magnitude (1.0 acelera uma view 100x100 em 100 pt/s²);

UIDynamicBehavior

- Provavelmente você irá querer fazer seu próprio **Behavior**...
- Isso se faz criando uma classe filha de **UIDynamicBehavior** e chamando o método:
 - - (**void**)addChildBehavior:(**UIDynamicBehavior** *)behavior;
- Dentro dos métodos sobrescritos:
 - **addItem** e **removeItem**

UIDynamicBehavior

- Todos os **UIDynamicBehavior** conhecem o **UIDynamicAnimator** do qual fazem parte;
- O UIDynamicBehavior tem a propriedade
 - **@property (copy) void (^action)(void);**
- A propriedade **action** é um bloco. Você pode definir para que ela faça qualquer coisa, mas faça um código eficiente, pois ele vai ser chamado **MUITAS VEZES** por segundo!

Hora de Brincar!

- Quadrados que explodem!
 - Código no GitHub (vamos refazer juntos)

Desafio

- Fazer os quadrados aparecerem na posição onde ocorrer um evento de tap;
- Fazer os quadrados quicarem quando tocarem o “chão”.