**AppFramework框架系统结构**

# 框架简介

AppFramework是淘宝自动化测试team自主研发的，解决iphone系统的应用UI自动化测试解决方案。

框架利用iphone应用本身一些特性，模拟用户的操作行为，完成对应用提供的功能自动化测试。

框架采用的是嵌入式测试思路，在应用的源码中嵌入测试代码，应用初始完成后生成测试对象，由测试对象完成自动化测试。

统一的控件管理、灵活的控件定位、简单的脚本编写是本框架特点。

# 系统结构

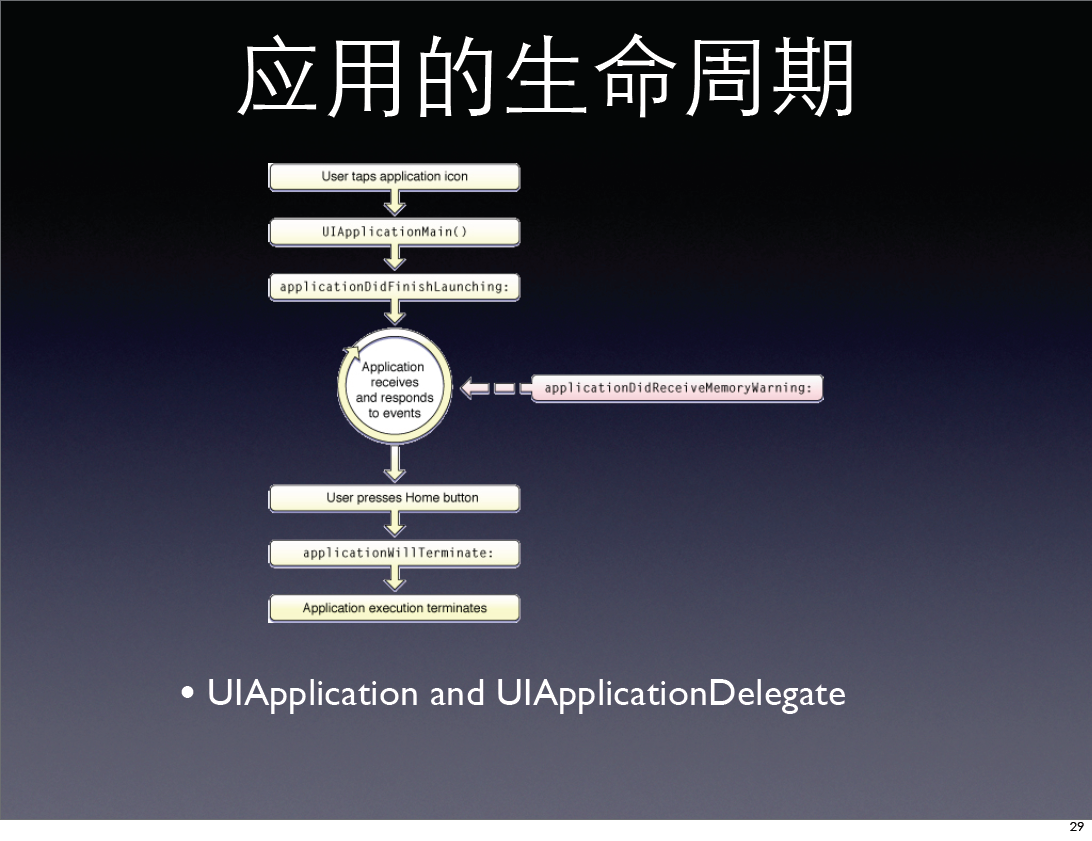
## Iphone应用特性

Iphone应用本身具备一些特性，下面对一些对自动化很有利的特性做简单说明：

1. 应用都是以window为基础，所有控件都内嵌在window内，控件之间也是一种内嵌方式，形成树状模式，非常方便控件的搜索
2. 在Iphone的应用中，所有视图控件的基类都是UIView，控件搜索和处理的时候也就变的相对灵活、简单了
3. UIView的superview和subview为控件树的遍历提供了条件
4. Touch事件在应用中被拆分为4个事件，touchesBegan、touchesMoved、touchesEnded、touchesCancelled

## 应用系统生命周期

下面的图诠释了iphone应用的整个生命周期。



框架在应用的初始过程中（applicationDidFinishlaunching）插入了负责执行测试的测试对象，在应用等待事件触发时，模拟了操作事件发送给应用进行处理。

## 框架结构

框架结构如下图所示：



框架由5个模块组成：TBTestRunner、TBElement、TBOperator、TBLog、TouchSynthesis。

**TBTestRunner**：该模块是测试对象，负责初始化设置，以及测试指令的解析

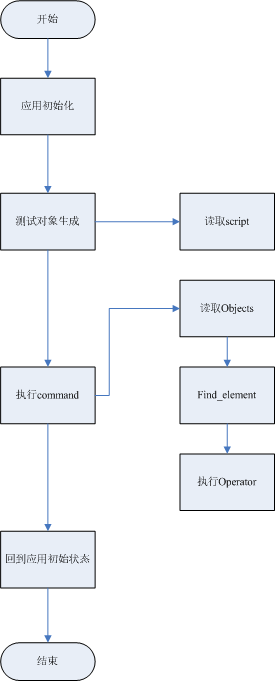
**TBElement**：该模块主要负责对操作控件的搜索，返回控件对象进行操作

**TBOperator**：该模块主要负责对操作指令的执行，通过TouchSynthesis模块的配合

**TouchSynthesis**：该模块主要负责对touch事件的模拟

## 框架机制

下图说明框架的工作机制：



* 在应用初始化时，生成测试对象
* 测试对象在初始化时，读取测试脚本信息
* 测试对象对测试脚本进行解析，执行里面的command
* 执行command时，搜索该command对应的控件
* 搜索完成后，在该控件上执行相应的操作
* 一条command执行完成后，执行下一条command
* 整个脚本执行完成后，释放测试对象