HUAWEI

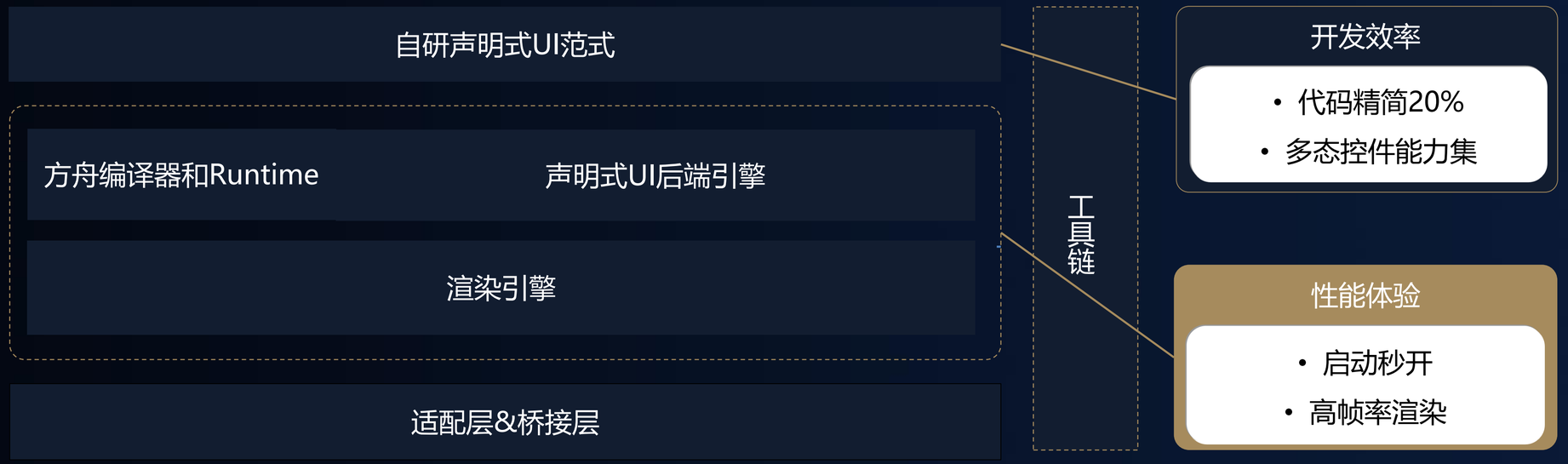
< HDC.Together >

HUAWEI DEVELOPER CONFERENCE 2021

HarmonyOS UI编程框架快速上手

HarmonyOS UI编程框架介绍

HarmonyOS UI（ArkUI）编程框架用于帮助开发者构建跨设备应用，在OS架构上属于上层框架。框架组成部分有：开发模型，声明式UI范式，系统API 。开发语言目前主要支持JS/TS语言。



声明式UI开发

UI = f(state)

The layout on the screen　Your build methods　The application state

用户界面任何状态的改变都只有一种编码途径。一旦给定任意状态，你就描述了用户界面应该长什么样，并且它就是这样。

命令式UI

```
// Imperative style
b.setColor(red)
b.clearChildren()
ViewC c3 = new ViewC(...)
b.add(c3)
```

声明式UI

```
// Declarative style
ViewB(color: red) {
   ViewC(...)
}
```

响应式数据绑定，UI逻辑分离，更直观，更高效，开发者只要关注核心数据即可

```
@Entry
@Component
struct Index {
  @State myText: string = 'World'

  build() {
    Column() {
      Text('Hello')
        .fontSize(100)
      Text(this.myText)
        .fontSize(100)
      Divider()
      Button() {
        Text('Click me')
      }.onClick(() => {
        this.myText = 'Huawei'
      })
    }
    .width('100%')
    .height('100%')
  }
}
```

1　2　3

代码示例

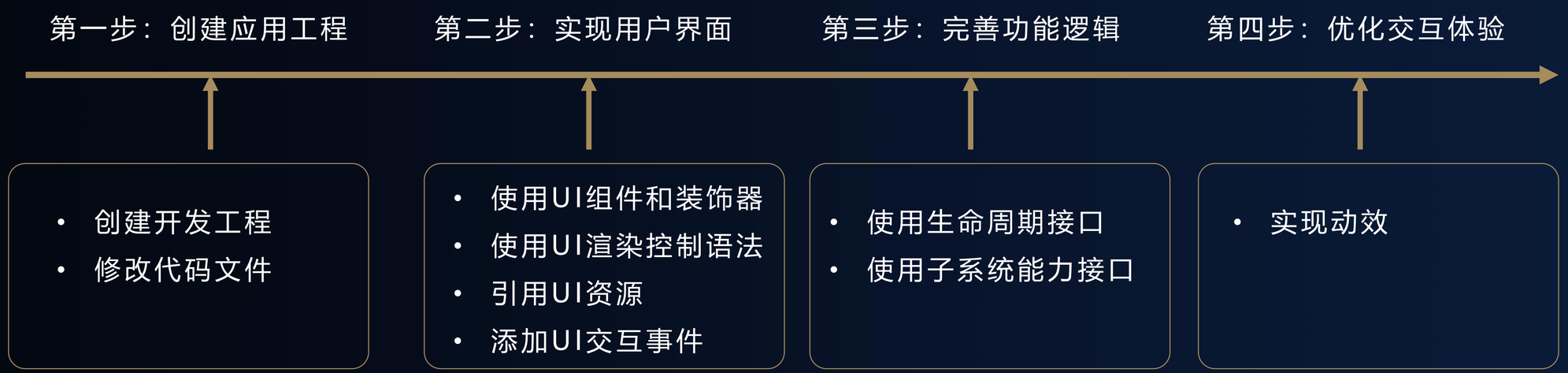
（1）结合装饰器定义布局

（2）进行UI布局描述

（3）添加事件方法

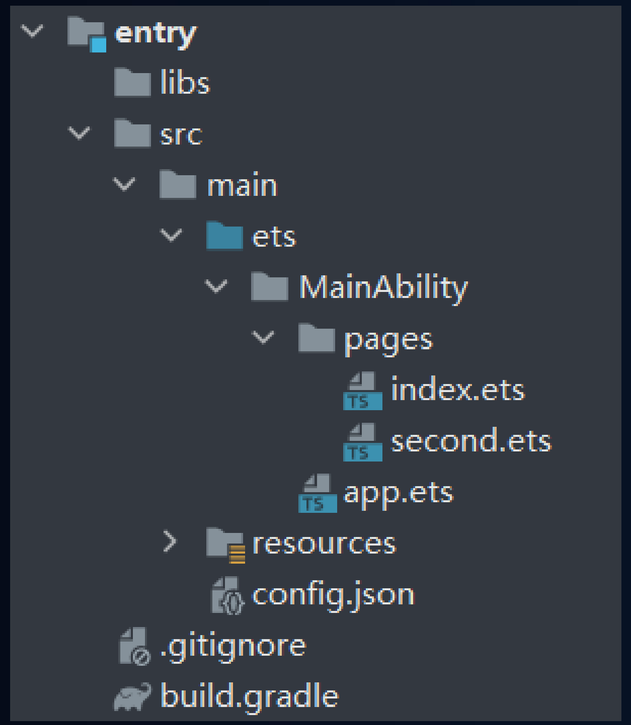
...

四步实现HarmonyOS应用



第一步创建应用工程

工程目录



代码文件

```
@Entry
@Component
struct Index {
  build() {
    Flex({
      direction: FlexDirection.Column,
      alignItems: ItemAlign.Center,
      justifyContent: FlexAlign.Center
    }) {
      Text('Hello HDC')
        .fontSize(50)
        .fontWeight(FontWeight.Bold)
      Button() {
        Text('next page')
          .fontSize(25)
          .fontWeight(FontWeight.Bold)
      }.type(ButtonType.Capsule)
      .margin({ top: 20 })
      .backgroundColor('#0D9FFB')
      .onClick(() => {
        routePage()
      })
    }
    .width('100%')
    .height('100%')
  }
}
```

第二步实现用户界面

使用UI组件和装饰器

基础组件：Image、Text、Video等
容器组件：Stack、Column、List等

实现组合目标面

组件化装饰：
@Component、@Entry、@Builder 、@Extend等

配套实现页面组件开发及组件自定义

状态管理装饰： @State 、@Link 、@Observed、@ObjectLink、@StorageLink、 @Watch

实现数据驱动视图自动更新

使用UI渲染控制语法

条件渲染：if/elseif/else

进行UI描述时，根据不同状态来动态控制组件的渲染

循环渲染：
ForEach/LazyForEach

进行UI描述时，根据数据的多少动态控制渲染的次数，优化代码实现

引用UI资源

字符串引用：
$r（‘app.string.name’）

resources的element目录下定义字符串，支持全球化小语种

媒体资源引用：
$r（‘app.media.name’）

resources的media目录下存放资源，支持png、jpg、svg等多种格式

添加UI交互事件

基础手势事件：onClick / onTouch等

定义基础用户交互，结合TouchEvent信息可以实现自定义手势

高级手势事件：长按手势 / 滑动手势 / 组合手势等

通过gesture属性函数配置内置高级手势支持，GestureGroup可支持多种高级手势组合

第三步完善功能逻辑

使用系统生命周期接口

页面生命周期接口：onPageShow、onPageHide
UI组件生命周期接口：aboutToAppear、aboutToDisappear
其它生命周期接口：onBackPressed、onCreate、onDestroy等

结合页面、UI组件、系统状态的变化生命周期接口添加功能逻辑

使用子系统能力接口

多个子系统提供大量系统能力接口
使用仅需两步：
一、导入依赖包。
二、直接调用系统能力接口。

调用系统能力实现具体功能逻辑

第四步优化交互体验

属性动画

属性动画animation ： 自动监听组件所有通用属性变化，自动增加动画补间

显式动画animateTo ： 指定特定属性变化，为特定的属性动画自动增加动画补间

修改组件属性，自动生成动画补间，优化属性变化交互体验

转场动画

组件间转场： transition监听组件的渲染状态变化，增加组件渲染、移除时的动画效果

页面间转场： pageTransition指定页面间跳转的切换动画效果

组件、页面切换时，自动生成动画补间，优化切换交互体验

动画组件

ImageAnimator：支持逐帧图片播放动画

使用多个图片组成动画，并动态控制动画播放

Animator：组件形式提供动画控制器

动态控制播放状态，定制补间动画，实现深度自定义动画效果

<HDC.Together>
华为开发者大会2021

扫码参加1024程序员节

<解锁HarmonyOS核心技能，赢取限量好礼>

开发者训练营

Codelabs 挑战赛

HarmonyOS技术征文

HarmonyOS开发者创新大赛

扫码了解1024更多信息

报名参加HarmonyOS
开发者创新大赛

谢谢

欢迎访问HarmonyOS开发者官网

欢迎关注HarmonyOS开发者微信公众号