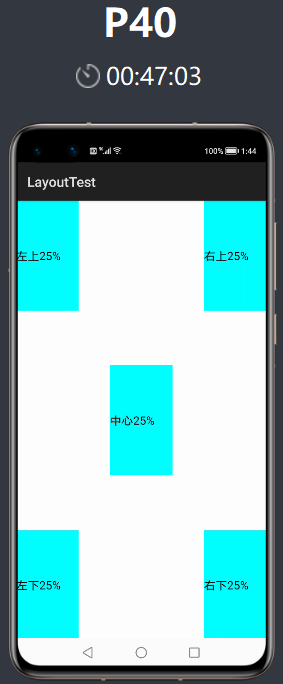
# PrecentPositionLayout功能介绍

## **组件介绍：**

鸿蒙SDK提供了不同布局规范的组件容器，例如以单一方向排列的DirectionalLayout、以相对位置排列的DependentLayout、以确切位置排列的PositionLayout等。

但是PositionLayout中组件的位置是以绝对像素点定义的，无法实现根据屏幕的大小自适应。因此，引入一种以百分比方式定义的PrecentPositionLayout布局容器，通过它可以很方便的实现屏幕自适应。

## **手机模拟器上运行效果：**



# **PrecentPositionLayout使用方法**

## **新建工程，增加组件Har包依赖**

在应用模块中调用HAR，常用的添加依赖的方式包括如下两种。

* 方式一：依赖本地HAR

只需要将precentpositionlayout.har复制到entry\libs目录下即可（由于build.gradle中已经依赖的libs目录下的\*.har，因此不需要在做修改）。

* 方式二：调用Maven仓中的HAR

在应用模块的build.gradle中的dependencies闭包中，添加如下代码。

dependencies **{** implementation **'com.huawei.har:precentpositionlayout:1.0.1'  
}**

## **修改主页面的布局文件**

修改主页面的布局文件ability\_main.xml，将跟组件容器修改为**com.isoftstone.precentpositionlayout.PrecentPositionLayout**，然后在增加5个Text组件，分别位于屏幕的左上，左下，右上，右下和中间的位置，长度和宽度都占屏幕的25%。修改后代码如下：

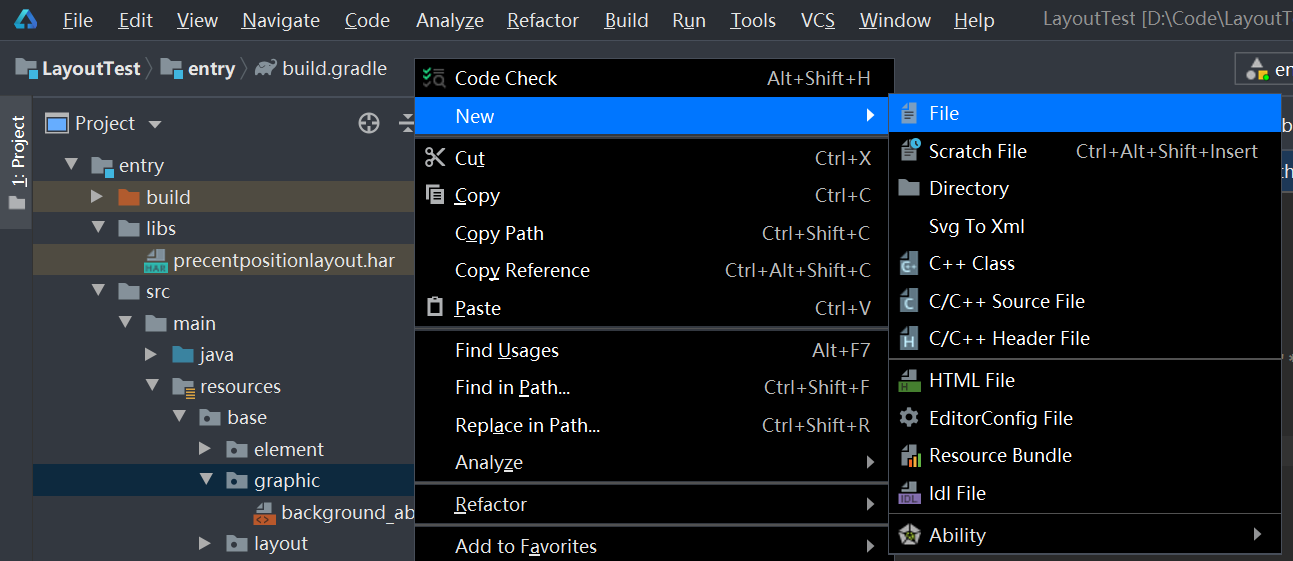
<?xml version**="1.0"** encoding**="utf-8"**?>  
  
<com.isoftstone.precentpositionlayout.PrecentPositionLayout  
 xmlns:ohos**="http://schemas.huawei.com/res/ohos"** ohos:height**="match\_parent"** ohos:width**="match\_parent"**>  
  
 <Text  
 ohos:id**="$+id:text\_helloworld"** ohos:height**="250"** ohos:width**="250"** ohos:left\_margin**="0"** ohos:top\_margin**="0"** ohos:background\_element**="$graphic:background\_text"** ohos:text**="左上25%"** ohos:text\_size**="50"** />  
  
 <Text  
 ohos:id**="$+id:text\_helloworld"** ohos:height**="250"** ohos:width**="250"** ohos:left\_margin**="750"** ohos:top\_margin**="0"** ohos:background\_element**="$graphic:background\_text"** ohos:text**="右上25%"** ohos:text\_size**="50"** />  
  
 <Text  
 ohos:id**="$+id:text\_helloworld"** ohos:height**="250"** ohos:width**="250"** ohos:left\_margin**="0"** ohos:top\_margin**="750"** ohos:background\_element**="$graphic:background\_text"** ohos:text**="左下25%"** ohos:text\_size**="50"** />  
  
 <Text  
 ohos:id**="$+id:text\_helloworld"** ohos:height**="250"** ohos:width**="250"** ohos:left\_margin**="750"** ohos:top\_margin**="750"** ohos:background\_element**="$graphic:background\_text"** ohos:text**="右下25%"** ohos:text\_size**="50"** />  
  
 <Text  
 ohos:id**="$+id:text\_helloworld"** ohos:height**="250"** ohos:width**="250"** ohos:left\_margin**="375"** ohos:top\_margin**="375"** ohos:background\_element**="$graphic:background\_text"** ohos:text**="中心25%"** ohos:text\_size**="50"** />  
  
</com.isoftstone.precentpositionlayout.PrecentPositionLayout>

## **增加Text组件的背景资源文件**

为了方便观察，上一步我们将Text组件设置了一个绘制背景graphic:background\_text。

这里需要在resources/base/grahic目录下新增一个可绘制资源文件。

右键点击graphic，选择New-File,文件名输入background\_text.xml。



文件内容如下：（可复制background\_ability\_main.xml的内容，修改color值即可）

<?xml version**="1.0"** encoding**="UTF-8"** ?>  
<shape xmlns:ohos**="http://schemas.huawei.com/res/ohos"** ohos:shape**="rectangle"**>  
 <solid  
 ohos:color**="#00FFFF"**/>  
</shape>

## **修改MainAbilitySlince的UI加载代码**

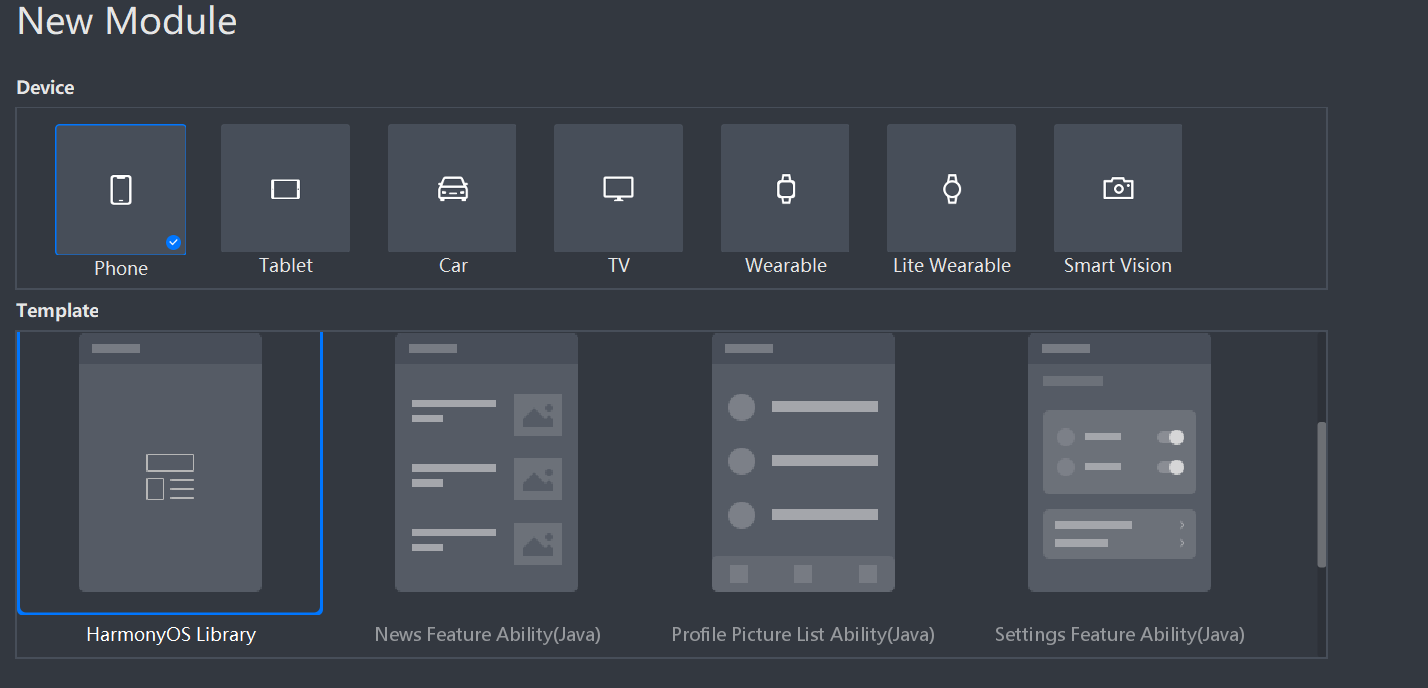
在MainAbilitySlince类的onStart函数中，增加如下代码。

public void onStart(Intent intent) {  
 super.onStart(intent);  
 // 解析xml获得PrecentPositionLayout对象  
 PrecentPositionLayout precentPositionLayout = (PrecentPositionLayout) LayoutScatter.*getInstance*(getContext()).parse(ResourceTable.*Layout\_ability\_main*, null, false);  
  
 // 自动调整组件的百分比  
 precentPositionLayout.AutoSize();  
  
 // 设置到UI  
 super.setUIContent(precentPositionLayout);  
 //super.setUIContent(ResourceTable.Layout\_ability\_main);  
}

# **PrecentPositionLayout开发实现**

## 新建一个Module

新建一个Module，类型选择HarmonyOS Library，模块名为precentpositionlayout，如图



## **新建一个PrecentPositionLayout类**

新建一个PrecentPositionLayout类，继承自PositionLayout类，并增加AutoSize()方法.

/\* 调整各组件的大小，按照百分比调整  
 \* 将原来组件的起始位置，宽度和高度都视作相对于整个屏幕的百分比值，然后根据屏幕的分辨率转换为实际的像素值。  
 \* 注：考虑到使用0-100配置百分比的话，范围太小不够精确，因此配置范围设置为0-1000，  
 \* 比如当前屏幕是1920 \* 1060, 某个组件的宽度和高度配置的是200，则表示改组件的宽和高都占整个屏幕的20%。  
 \* 因此，调整后改组件的实际大小为384 \* 212.  
 \*/  
public void AutoSize() {  
 // 获取屏幕分辨率  
 Optional<Display> display = DisplayManager.*getInstance*().getDefaultDisplay(this.getContext());  
 Point pt = new Point();  
 display.get().getSize(pt);  
  
 // 去除上面标题栏和下面导航栏的高度  
 pt.modify(pt.getPointX(), pt.getPointY() - 160);  
  
 // 调增各组件的大小  
 int childCount = getChildCount();  
 for (int i = 0; i < childCount; i++) {  
 Component component = getComponentAt(i);  
 ComponentContainer.LayoutConfig config = component.getLayoutConfig();  
 component.setLeft(config.getMarginLeft() \* pt.getPointXToInt() / 1000);  
 component.setTop(config.getMarginTop() \* pt.getPointYToInt() / 1000);  
 component.setWidth(config.width \* pt.getPointXToInt() / 1000);  
 component.setHeight(config.height \* pt.getPointYToInt() / 1000);  
 }  
}

## **编译HAR包**

利用Gradle可以将HarmonyOS Library库模块构建为HAR包，构建HAR包的方法如下：

在Gradle构建任务中，双击PackageDebugHar或PackageReleaseHar任务，构建Debug类型或Release类型的HAR。

待构建任务完成后，可以在工程目录中的PrecentPositionLayout> bulid > outputs > har目录中，获取生成的HAR包。

