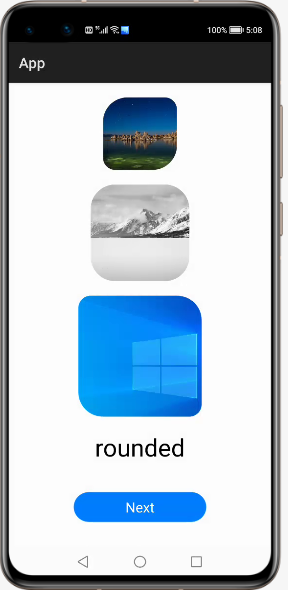
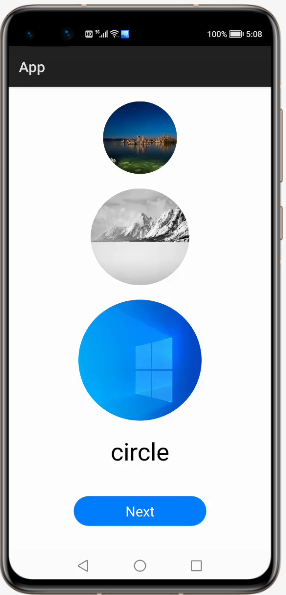
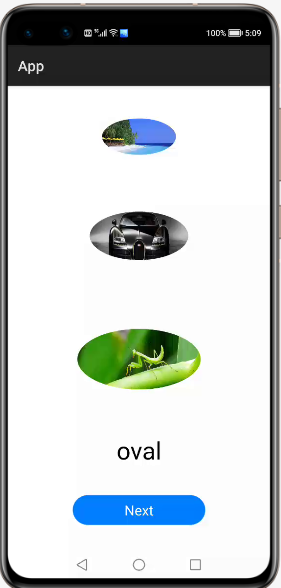
# RoundedImage组件功能介绍

## 功能介绍：

RoundedImage组件是一个可以将图片显示成圆形，椭圆形，圆角矩形的，目前仅支持三种样式的显示。

## 模拟器上运行效果：

# RoundedImage**使用方法**

## 新建工程，增加组件Har包依赖

在应用模块中添加HAR，只需要将library-debug.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）。

## **修改主页面的布局文件**

修改主页面的布局文件ability\_main.xml，增加**com.custom.library.RoundedImage**组件，组件的宽和高自定义。

*<?***xml version="1.0" encoding="utf-8"***?>*<**DirectionalLayout  
 xmlns:ohos="http://schemas.huawei.com/res/ohos"  
 ohos:height="match\_parent"  
 ohos:width="match\_parent"  
 ohos:orientation="vertical"**>  
  
 <**com.custom.library.RoundedImage  
 ohos:id="$+id:image1"  
 ohos:height="300"  
 ohos:width="300"  
 ohos:top\_margin="20vp"  
 ohos:layout\_alignment="center"**/>  
 <**com.custom.library.RoundedImage  
 ohos:id="$+id:image2"  
 ohos:height="400"  
 ohos:width="400"  
 ohos:layout\_alignment="center"  
 ohos:top\_margin="20vp"**/>  
 <**com.custom.library.RoundedImage  
 ohos:id="$+id:image3"  
 ohos:height="500"  
 ohos:width="500"  
 ohos:layout\_alignment="center"  
 ohos:top\_margin="20vp"**/>  
</**DirectionalLayout**>

## **修改MainAbilitySlince的UI加载代码**

在MainAbilitySlince类的onStart函数中。

***增加如下代码可显示圆角矩形：***

@Override  
**public void** onStart(Intent intent) {  
 **super**.onStart(intent);  
 **super**.setUIContent(ResourceTable.***Layout\_ability\_main***);  
  
 RoundedImage roundedImage1 = (RoundedImage) findComponentById(ResourceTable.***Id\_image1***);  
 roundedImage1.setPixelMapToRoundedRect(ResourceTable.***Media\_photo***, 100, 50, 100, 50);  
 RoundedImage roundedImage2 = (RoundedImage) findComponentById(ResourceTable.***Id\_image2***);  
 roundedImage2.setPixelMapToRoundedRect(ResourceTable.***Media\_photo1***, 100, 100, 100, 100);  
 RoundedImage roundedImage3 = (RoundedImage) findComponentById(ResourceTable.***Id\_image3***);  
 roundedImage3.setPixelMapToRoundedRect(ResourceTable.***Media\_photo2***, 50, 100, 50, 100);  
 }

***增加如下代码可显示圆形***：

@Override  
**public void** onStart(Intent intent) {  
 **super**.onStart(intent);  
 **super**.setUIContent(ResourceTable.***Layout\_ability\_main***);  
  
 RoundedImage roundedImage1 = (RoundedImage) findComponentById(ResourceTable.***Id\_image1***);  
 roundedImage1.setPixelMapToCircleImage(ResourceTable.***Media\_photo***);  
  
 RoundedImage roundedImage2 = (RoundedImage) findComponentById(ResourceTable.***Id\_image2***);  
 roundedImage2.setPixelMapToCircleImage(ResourceTable.***Media\_photo1***);  
  
 RoundedImage roundedImage3 = (RoundedImage) findComponentById(ResourceTable.***Id\_image3***);  
 roundedImage3.setPixelMapToCircleImage(ResourceTable.***Media\_photo2***);  
}

***增加如下代码可显示椭圆形：***

@Override  
**public void** onStart(Intent intent) {  
 **super**.onStart(intent);  
 **super**.setUIContent(ResourceTable.***Layout\_ability\_main***);  
  
 RoundedImage roundedImage1 = (RoundedImage) findComponentById(ResourceTable.***Id\_image1***);  
 roundedImage1.setPixelMapToOvalImage(ResourceTable.***Media\_photo3***);  
  
 RoundedImage roundedImage2 = (RoundedImage) findComponentById(ResourceTable.***Id\_image2***);  
 roundedImage2.setPixelMapToOvalImage(ResourceTable.***Media\_photo4***);  
  
 RoundedImage roundedImage3 = (RoundedImage) findComponentById(ResourceTable.***Id\_image3***);  
 roundedImage3.setPixelMapToOvalImage(ResourceTable.***Media\_photo5***);  
}

# RoundedImage**开发实现**

## **新建一个Module**

新建一个Module，类型选择HarmonyOS Library，模块名为library。

## **新建一个RoundedImage类**

新建一个RoundedImage类，继承自Image类，实现DrawTask.onDraw接口，代码如下：

*用于绘制圆形：*

@Override  
**public void** onDraw(Component component, Canvas canvas) {  
 **float** centerX = getWidth() / 2f;  
 **float** centerY = getHeight() / 2f;  
 **float** radius = Math.*min*(centerX, centerY);  
 Paint paint = **new** Paint();  
 Shader shader = **new** PixelMapShader(**holder**, Shader.TileMode.***CLAMP\_TILEMODE***, Shader.TileMode.***CLAMP\_TILEMODE***);  
 paint.setShader(shader, Paint.ShaderType.***SWEEP\_SHADER***);  
 canvas.drawCircle(centerX, centerY, radius, paint);  
}

*用于绘制椭圆形：*

@Override  
**public void** onDraw(Component component, Canvas canvas) {  
 Paint paint = **new** Paint();  
 Shader shader = **new** PixelMapShader(**holder**, Shader.TileMode.***CLAMP\_TILEMODE***, Shader.TileMode.***CLAMP\_TILEMODE***);  
 paint.setShader(shader, Paint.ShaderType.***SWEEP\_SHADER***);  
 PixelMap pixelMap = **holder**.getPixelMap();  
 **int** min = Math.*min*(pixelMap.getImageInfo().**size**.**width**, pixelMap.getImageInfo().**size**.**height**);  
 **int** radiusX = Math.*min*(min, **minImageLength**);  
 **float** halfRadiusX = radiusX / 2f;  
 **float** quarterRadiusX = radiusX / 4f;  
 **float** left = getWidth() / 2f - halfRadiusX;  
 **float** right = getWidth() / 2f + halfRadiusX;  
 **float** top = getHeight() / 2f - quarterRadiusX;  
 **float** bottom = getHeight() / 2f + quarterRadiusX;  
 RectFloat rect = **new** RectFloat(left, top, right, bottom);  
 canvas.drawOval(rect, paint);  
}

*用于设置圆角矩形，调用Image方法进行设置：*

setCornerRadii(**new float**[]{topLeft, topLeft, topRigth, topRigth, bottomRight, bottomRight, bottomLeft, bottomLeft});

## **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loadingview> bulid > outputs > har目录中，获取生成的HAR包。