

自定义布局

Egret eui库内置的4个布局类基本上可以满足大部分需求，不过在一些特殊情况下，可能会需要一些特殊的布局方式。此时可以制作自定义布局类，代替默认的布局类。

新建一个MyLayout.ts，扩展自 eui.LayoutBase:

```

module uilayout {
    var UIComponentClass = "eui.UIComponent";
    /** 自定义的环形布局类 */
    export class RingLayout extends eui.LayoutBase{
        public constructor(){
            super();
        }
        /**
         * 计算target的尺寸
         * 因为环形布局，依赖容器尺寸来定义半径，所以需要容器显式的设置width和height，在这
            种情况下measure方法将失去作用
         * 所以在这个例子里面，不需要重写measure方法
         * 如果您的自定义布局需要根据内部子项计算尺寸，请重写这个方法
         */
        public measure():void{
            super.measure();
        }
        /**
         * 重写显示列表更新
         */
        public updateDisplayList(unscaledWidth:number, unscaledHeight:number):void{
            super.updateDisplayList(unscaledWidth, unscaledHeight);
            if (this.target==null)
                return;
            var centerX:number = unscaledWidth/2;// 获得容器中心的X坐标
            var centerY:number = unscaledHeight/2;// 获得容器中心的Y坐标
            var horizon:number = centerX/2;// 获得水平可用长度的一半
            var vertical:number = centerY/2;// 获得垂直可用长度的一半
            var radius = horizon > vertical ? vertical : horizon;// 取小的为圆形半径
            var count:number = this.target.numElements;
            var maxX:number = 0;
            var maxY:number = 0;
            for (var i:number = 0; i < count; i++){
                var layoutElement:eui.UIComponent = <eui.UIComponent> ( this.target.
getElementAt(i) );
                if ( !egret.is(layoutElement,UIComponentClass) || !layoutElement.inc
ludeInLayout ) {
                    continue;
                }
                var elementWidth:number = 0;
                var elementHeight:number = 0;
            }
        }
    }
}

```

```

        var angle:number = 2 * Math.PI * i / count;// 获得角度的大小
        var childX:number = centerX + radius * Math.sin(angle) - elementWidth/2;// 获得圆周点的X坐标
        var childY:number = centerY - radius * Math.cos(angle) - elementHeight/2;// 获得圆周点的Y坐标
        layoutElement.setLayoutBoundsPosition(childX, childY);
        maxX = Math.max(maxX,childX+elementWidth);
        maxY = Math.max(maxY,childY+elementHeight);
    }
    this.target.setContentSize(maxX,maxY);
}
}
}

```

然后修改前几节中的例子，改为使用自定义布局类(通过设置group.layout属性)：

基本代码可以参考前面几节。下面是修改其中的myGroup为自定义布局。

```

this.myGroup.horizontalCenter = 0;
this.myGroup.layout = new uilayout.RingLayout();//自定义布局
this.addChild( this.myGroup );
//内部对象
for(var i:number=0;i<20;i++) {
    var btn:eui.Button = new eui.Button();
    btn.width = 40;
    btn.height = 40;
    btn.label = "Button"+i;
    this.myGroup.addChild( btn );
}

```

运行生成效果：

