Go语言的移动端应用

Go移动端的起源

- 1、Go自1.4版本起，开始支持移动端开发(仅限于Android)。
- 2、1.5版本开始支持ios。
- 3、1.6版部分优化。

目录

gomobile工具

第一步：

获取工具

go get golang.org/x/mobile/cmd/gomobile

(*注意)

```
➜  src  go get golang.org/x/mobile/cmd/gomobile
go install golang.org/x/mobile/cmd/gomobile: open /usr/local/go/bin/gomobile: permission denied
➜  src
```

(当前窗口不会立即生效,需重新打开命令行)

第二步：

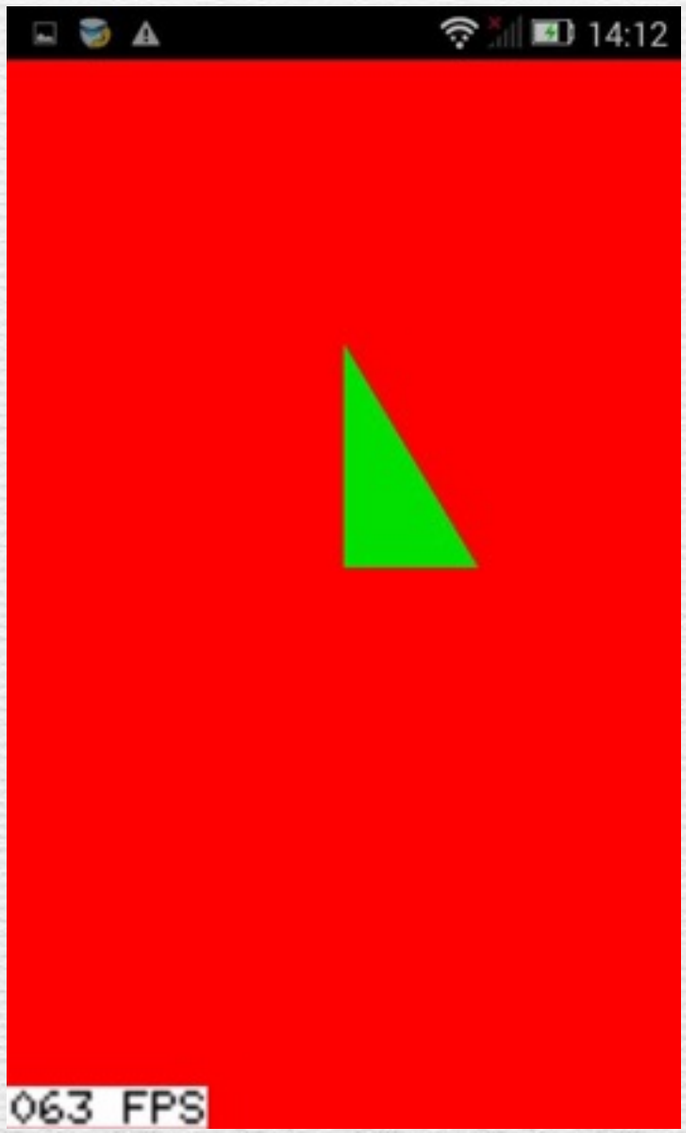
gomobile init

(从google下载ndk,需翻墙)

测试gomobile是否安装成功

gomobile build -target=android golang.org/x/mobile/example/basic

(如果成功，会生成一个basic.apk的android安装包)



最基础的包 app

1、Main()

调用并运行移动应用

2、类型 App

开发移动应用的基础

App的主要方法

1、Events()　//事件返回的通道

1) lifecycle.Event　//生命周期

2) mouse.Event　　//鼠标

3) paint.Event　　//渲染

4) size.Event　　//尺寸

5) touch.Event　　//触摸

2、Send()　//发送一个事件到事件通道

3、Publish()　//发布所有的操作命令

4、Filter()　　//注册事件过滤器

5、RegisterFilter()　//注册事件

程序运行的基本结构

```go
package main

import (
    "golang.org/x/mobile/app"
    "golang.org/x/mobile/event/lifecycle"
)

func main() {
    app.Main(func(a app.App) {
        // 遍历app的事件
        for e := range a.Events() {
            // 选择判断事件类型
            switch e := a.Filter(e).(type) {
            //是否是生命周期类型
            case lifecycle.Event:
                //判断app窗口是否可见
                switch e.Crosses(lifecycle.StageVisible) {
                case lifecycle.CrossOn:
                    // code...
                case lifecycle.CrossOff:
                    // code...
                }
            }
        }
    })
}

```

有点变化

1、获取app环境

2、设置背景颜色

3、发布显示

```
23                  case lifecycle.CrossOn:
24                      var glctx gl.Context
25                      // 获取app环境
26                      glctx, _ = e.DrawContext.(gl.Context)
27                      // 设置颜色
28                      glctx.ClearColor(0.51, 0, 0.51, 1)
29                      glctx.Clear(gl.COLOR_BUFFER_BIT)
30                      // 显示发布
31                      a.Publish()
32
33                  case lifecycle.CrossOff:
```

事件

lifecycle包　//管理生命周期

三个阶段：

1、CrossNone　//无

2、CrossOn　　//显示

3、CrossNone　//隐藏

事件

mouse包　//管理鼠标

类型Button

ButtonNone　　//无

ButtonLeft　　//左键

ButtonMiddle　　//按下滚轮

ButtonRight　　//右键

ButtonWheelUp　　//向上滚动

ButtonWheelDown　　//向下滚动

事件

mouse包　//管理鼠标

类型Direction

DirNone　　//无

DirPress　　//按下

DirRelease　　//抬起

事件

size包　//管理尺寸、物理分辨率、方向

类型 Event

WidthPx, HeightPx int　　//屏幕像素大小

WidthPt, HeightPt geom.Pt　//屏幕物理大小

PixelsPerPt float32　　//屏幕分辨率

Orientation Orientation　　//屏幕显示方向

事件

touch包　//触碰操作

类型Event

X,Y　//坐标点

Sequence　//序列

类型Type　//触碰

类型Type

TypeBegin　//开始触碰

TypeMove　//触碰中移动

TypeEnd　//结束触碰

事件

key包　//键盘

　　类型 Code　　//键码

类型Direction

DirNone	//无
DirPress	//按下
DirRelease	//抬起

网络

1、网络访问

Android需要AndroidManifest.xml 文件中开启网络权限

快捷通道

exp/app/debug	Package debug provides GL-based debugging tools for apps.
exp/audio	Package audio provides a basic audio player.
exp/audio/al	Package al provides OpenAL Soft bindings for Go.
exp/f32	Package f32 implements some linear algebra and GL helpers for float32s.
exp/font	Package font provides platform independent access to system fonts.
exp/gl/glutil	Package glutil implements OpenGL utility functions.
exp/sensor	Package sensor provides sensor events from various movement sensors.
exp/sprite	Package sprite provides a 2D scene graph for rendering and animation.
exp/sprite/clock	Package clock provides a clock and time functions for a sprite engine.
exp/sprite/glsprite	Package glsprite implements a sprite Engine using OpenGL ES 2.
exp/sprite/portable	Package portable implements a sprite Engine using the image package.