

现代操作系统应用开发实验报告

学号： 14331006

班级： 教务二班

姓名： 毕心然

实验名称： HW14

一 . 参考资料

<http://blog.csdn.net/gf771115/article/details/24605155>

<http://www.cocoachina.com/cocos/20140510/8369.html>

二 . 实验步骤

【开发平台为 win10】

- 1、按照要求配置 JAVA 环境
- 2、在 demo 基础上按照课件中讲的方法添加并完成按钮的回调函数
- 3、完成 onHttpRequestScoreCompleted 等函数

三 . 实验结果截图

```
C:\WINDOWS\system32\cmd.exe - java -jar server.jar

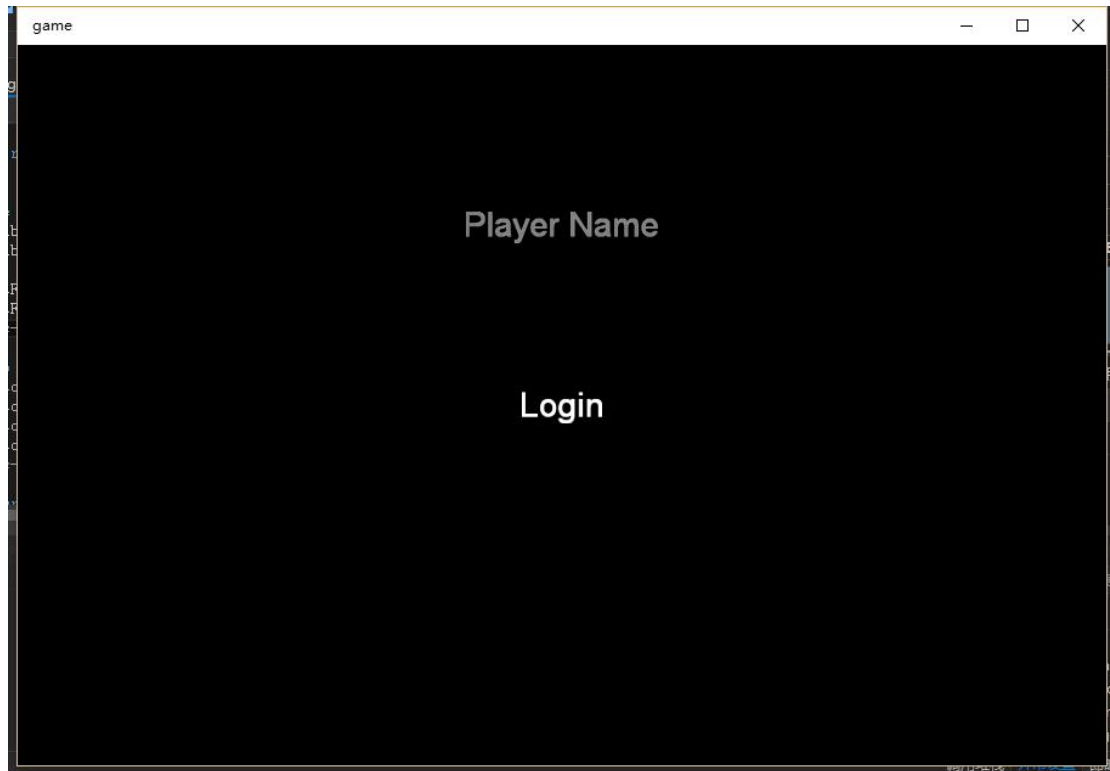
G:\anna>cd G:\anna\coding\cocos\week14
G:\anna>java -jar server.jar

=====
:: Spring Boot :: (v1.3.3.RELEASE)

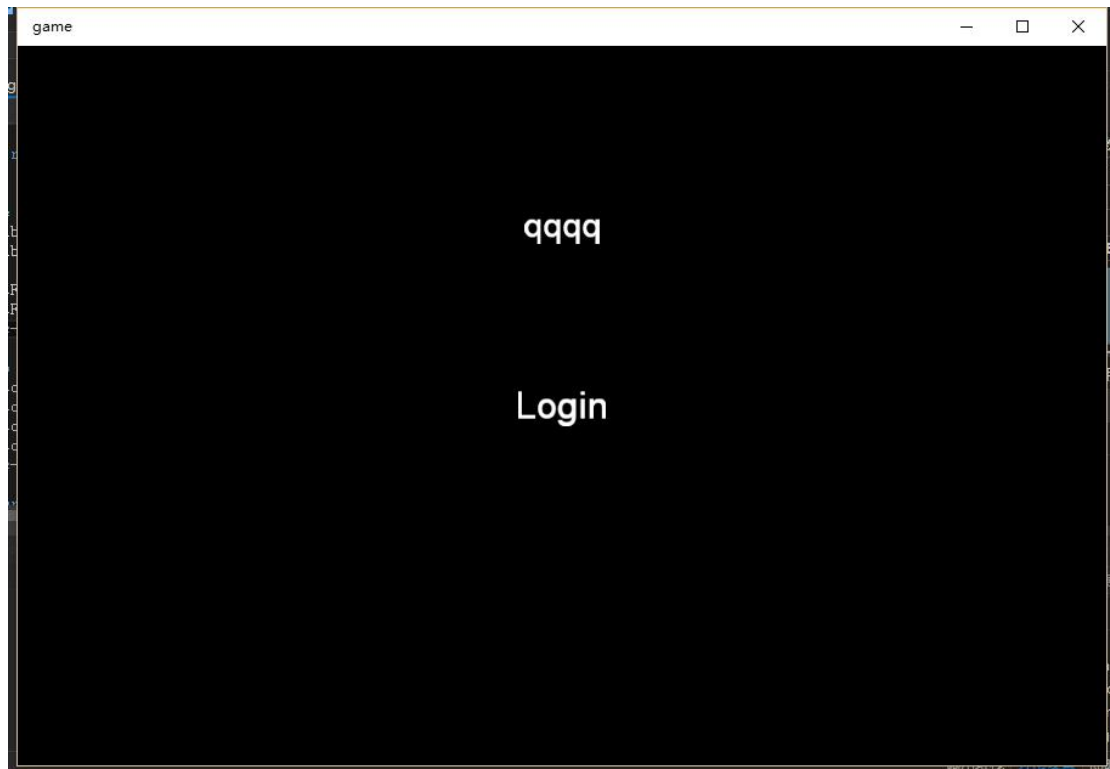
2016-05-30 15:51:18.323 INFO 11700 --- [main] hello.Application : Starting Application
2016-05-30 15:51:18.327 INFO 11700 --- [main] hello.Application : No active profile set, falling back to default profiles: default
2016-05-30 15:51:18.394 INFO 11700 --- [main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@1153f2b: startup date [Mon May 30 15:51:18 CST 2016]; root of context hierarchy
2016-05-30 15:51:18.995 INFO 11700 --- [main] o.s.b.f.s.DefaultListableBeanFactory : Overriding bean definition for bean 'beanNameViewResolver' with a different definition: replacing [Root bean: class [null], scope=, abstract=true, lazyInit=false, autowireMode=3, dependencyCheck=0, autowireCandidate=true, primary=false, factoryBeanName=org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WhitelabelErrorViewConfiguration, factoryMethodName=beanNameViewResolver; initMethodName=null; destroyMethodName=(inferred); defined in class path resource [org/springframework/boot/autoconfigure/web/ErrorMvcAutoConfiguration$WhitelabelErrorViewConfiguration.class]] with [Root bean: class [null], scope=, abstract=false, lazyInit=false, autowireMode=3, dependencyCheck=0, autowireCandidate=true, primary=false, factoryBeanName=org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfigurationAdapter, factoryMethodName=beanNameViewResolver; initMethodName=null; destroyMethodName=(inferred); defined in class path resource [org/springframework/boot/autoconfigure/web/WebMvcAutoConfiguration$WebMvcAutoConfigurationAdapter.class]]

2016-05-30 15:51:22.560 INFO 11700 --- [main] s.w.s.m.m.a.RequestMappingMapping : Mapped " [/error], produces=[text/html]" onto public org.springframework.web.servlet.ModelAndView org.springframework.boot.autoconfigure.web.BasicErrorController.errorHtml(javax.servlet.http.HttpServletRequest)
2016-05-30 15:51:22.669 INFO 11700 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2016-05-30 15:51:22.759 INFO 11700 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/*] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2016-05-30 15:51:22.952 INFO 11700 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans found on JMX exposure on startup
2016-05-30 15:51:23.272 INFO 11700 --- [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http)
2016-05-30 15:52:33.272 INFO 11700 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Started Application in 5.318 seconds (JVM running for 6.021)
2016-05-30 15:52:33.226 INFO 11700 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring FrameworkServlet 'dispatcherServlet'
2016-05-30 15:52:33.228 INFO 11700 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started
2016-05-30 15:52:33.571 INFO 11700 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 341 ms
```

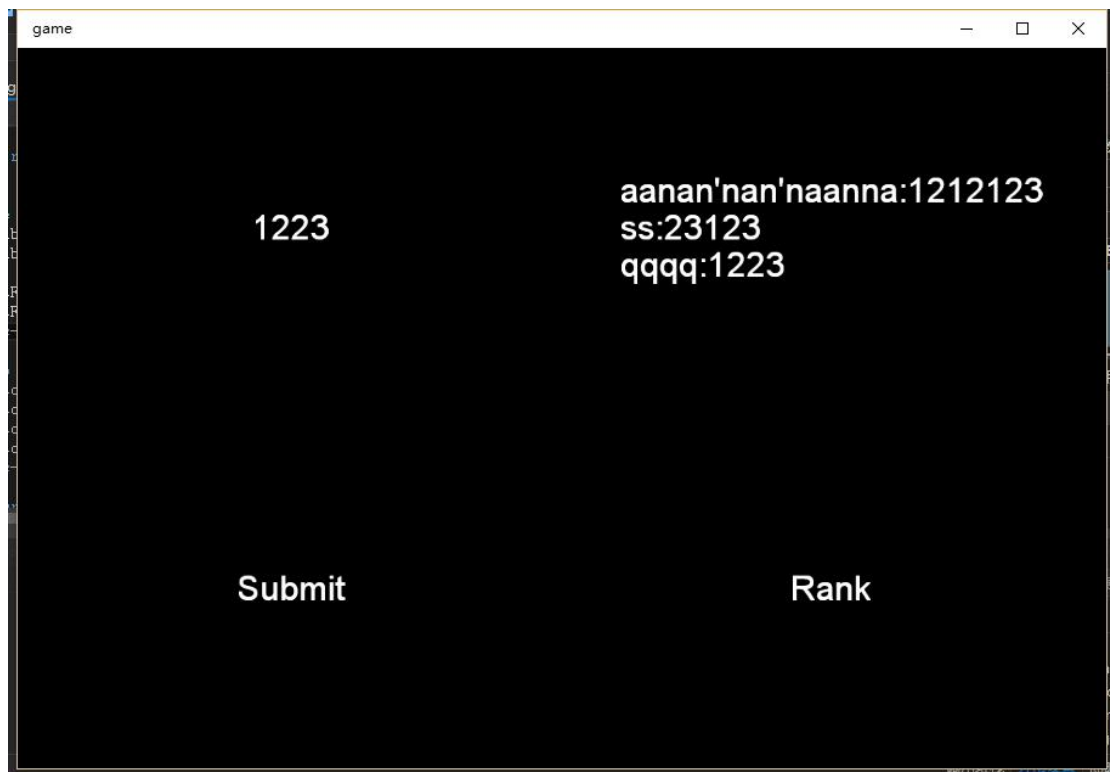
2、初始页面



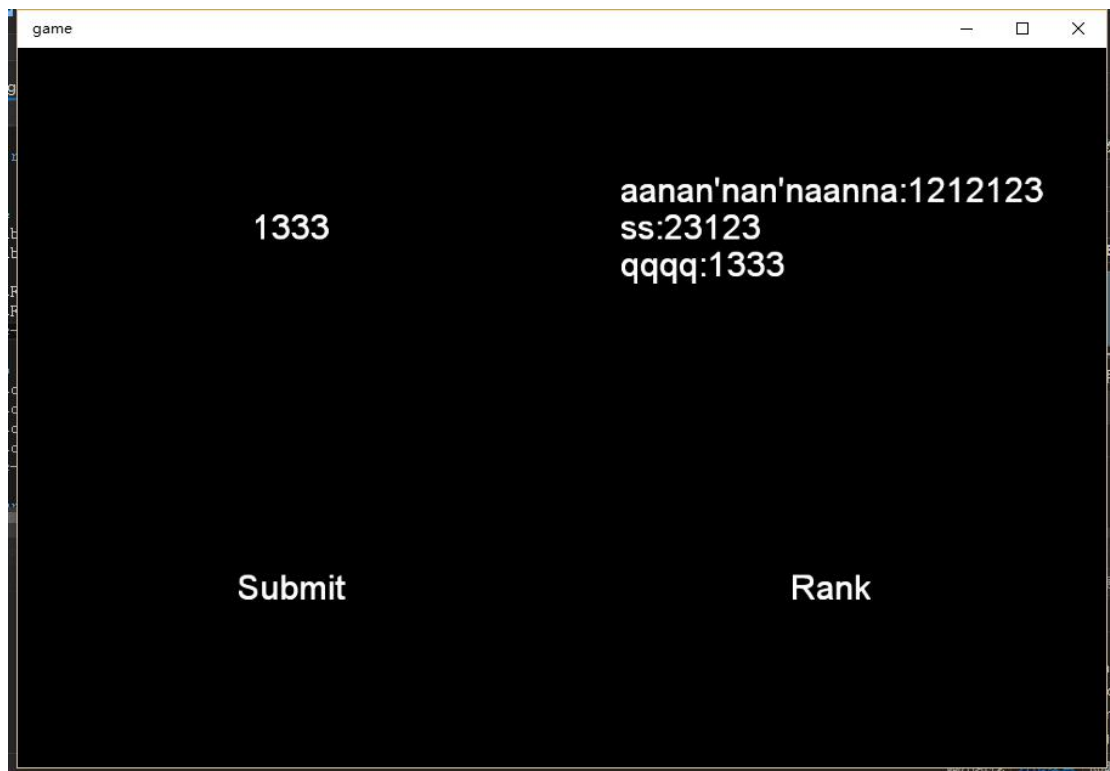
3、输入用户名后

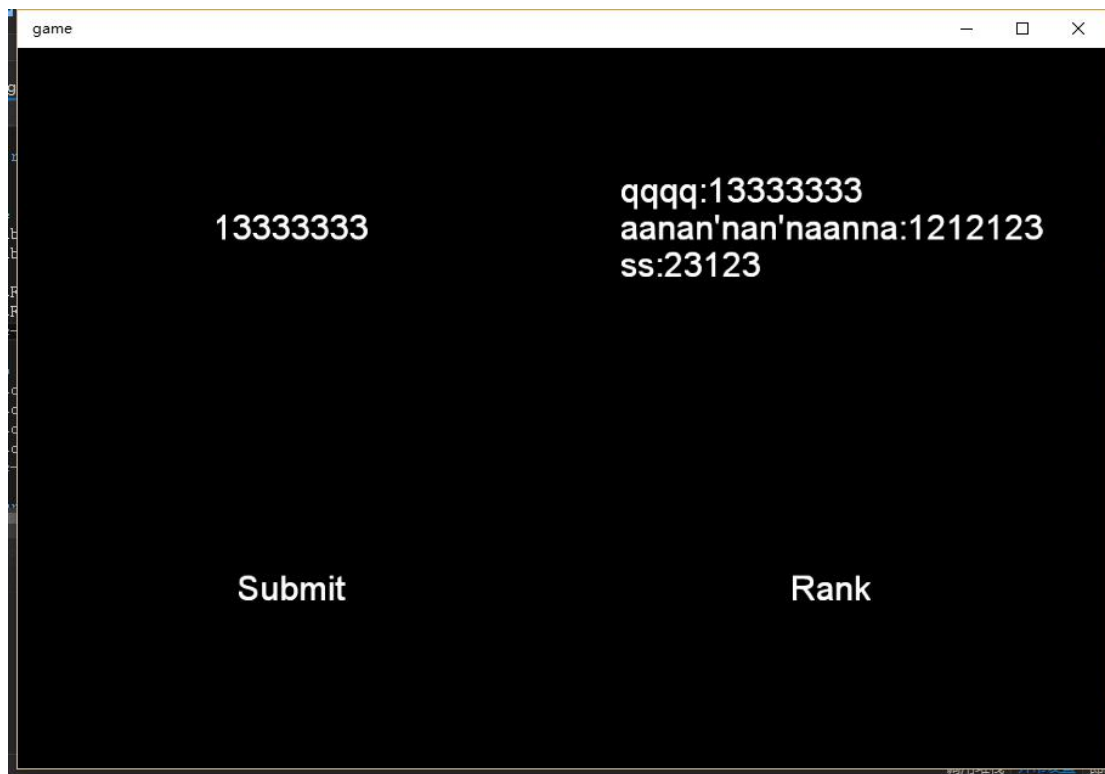


4、跳转，输入分数，点击 submit 后点击 rank 获取排名



5、修改分数重新提交和获取排名





6、rank 的点击事件响应函数部分代码

```
void GameScene::RankClick(Ref *pSender, Widget::TouchEvent type) {
    if (type == Widget::TouchEvent::ENDED) {
        HttpRequest* request = new HttpRequest();
        std::stringstream ss;
        ss << rand();
        string s;
        ss >> s;
        request->setUrl(("http://localhost:8080/rank?top=10&rand="+s).data());
        request->setRequestType(HttpRequest::Type::GET);
        request->setResponseCallback(CC_CALLBACK_2(GameScene::onHttpRequestRankCompleted, this));
        request->setTag("GET test");

        vector<string> headers;
        headers.push_back("Cookies: GAMESESSIONID=" + Global::getSessionIdFromHeader(Global::gameSessionId));
        headers.push_back("Content-Type: application/x-www-form-urlencoded; charset=UTF-8");
        request->setHeaders(headers);

        cocos2d::network::HttpClient::getInstance()->send(request);
        request->release();
    }
}
```

7、submit 的点击事件响应函数部分代码

```
69
70 void GameScene::SubmitClick(Ref *pSender, Widget::TouchEventType type) {
71     if (type == Widget::TouchEventType::ENDED) {
72         string score = score_field->getString();
73         HttpRequest* request = new HttpRequest();
74         request->setRequestType(HttpRequest::Type::POST);
75         request->setUrl("http://localhost:8080/submit");
76         request->setResponseCallback(CC_CALLBACK_2(GameScene::onHttpRequestScoreCompleted, this));
77         string data = "score=" + score;
78         const char* postData = data.data();
79         request->setRequestData(postData, strlen(postData));
80         request->setTag("POST test");
81
82         vector<string> headers;
83         headers.push_back("Cookies: GAMESESSIONID="+Global::getSessionIdFromHeader(Global::gameSessionId));
84         headers.push_back("Content-Type: application/x-www-form-urlencoded; charset=UTF-8");
85         request->setHeaders(headers);
86
87         cocos2d::network::HttpClient::getInstance()->send(request);
88         request->release();
89     }
90 }
```

四 . 验过程遇到的问题

1、最初下载的 demo 遇到了许多奇奇怪怪的问题...群上也有许多人遇到同样的问题，有些看到解决方案的就跟着改了一些。后来看到 TA 上传了新的 demo 就重新用 proj.win10 开始了。

2、按照课件完成代码后 textblock 一直无法输入...后来意识到不会出现光标 ,直接输入就好.....

五 . 思考与总结

其实每个内容学到最后需要 http 请求的时候我都挺头痛的，最开始是 web，那时候有一次作业是服务器的返回和客户端的请求都要自己写，不能用现成的框架之类。虽然对整个的流程都有了比较清晰的概念但那次的痛苦还是有着隐隐的阴影....不过对后面的课程确实挺有帮助，现在调用各种函数感觉理解起来并不难，但仍需好好熟悉与联系才能掌握好。