

现代操作系统应用开发实验报告

学号： 15331248

班级： 上午班

姓名： 潘承远

实验名称： Cocos2d(HW15)

一.参考资料

- http://blog.csdn.net/nanshan_hzq/article/details/39965665 了解 cocos2d-x httpclient 发送 get 或者 post 消息请求
- http://www.cnblogs.com/hongten/p/hongten_get_post.html get 和 post 方法的区别
- 《网络 and 常用算法》这是老师给的 pdf, 了解了网络的基本概念、如何使用 HttpClient 等知识
- <http://www.tuicool.com/articles/MnuAVb> 了解 cmd 有关 jar 的几个命令
- <https://zhidao.baidu.com/question/1742817.html> 复习 url 的定义

二.实验步骤

- 仔细阅读课件与相关文档
- 配置 java 开发环境, 使用 java 命令运行服务端程序
- 完成基本要求: “实现使用用户名登录的功能”、“在界面呈现返回的 Header 和 Body 信息”

下面是关键代码: 第一张代码是 button 的点击事件函数, 第二张是 HttpRequest 的回调函数

```
void LoginScene::startMenuCallback() {
    HttpRequest* request = new HttpRequest();
    request->setUrl("http://localhost:8080/login");
    request->setRequestType(HttpRequest::Type::POST);
    request->setResponseCallback(CC_CALLBACK_2(LoginScene::onHttpRequestCompleted, this));

    // write the post data
    string str = "username=" + textField->getString();
    const char* postData = str.c_str();
    request->setRequestData(postData, strlen(postData));
    request->setTag("POST login");

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

    //auto scene = GameScene::createScene();
    //Director::getInstance()->replaceScene(TransitionFade::create(2, scene));
}
```

```
void LoginScene::onHttpRequestCompleted(HttpClient *sender, HttpResponse *response)
{
    if (!response)
    {
        return;
    }
    if (!response->isSucceed())
    {
        log("response failed");
        log("error buffer: %s", response->getErrorBuffer());
        return;
    }
    // dump data
    std::vector<char> *dataBuffer = response->getResponseData();
    std::vector<char> *headerBuffer = response->getResponseHeader();
    string datatemp = Global::toString(dataBuffer);
    string headertemp = Global::toString(headerBuffer);
    string temp = "";
    temp += headertemp;
    temp += '\n';
    temp += datatemp;
    textField->setText(temp);
    //Global::gameSessionId = Global::getSessionIdFromHeader(headertemp);
}
```

- 完成冲击高分：“实现使用用户名登录的功能”、“实现提交分数的功能”、“实现查询最好的 n 位成绩的功能”

下面是完成本部分的关键代码

这是 submit_button 的点击事件函数

```
void GameScene::submit() {
    HttpRequest* request = new HttpRequest();
    request->setUrl("http://localhost:8080/submit");
    request->setRequestType(HttpRequest::Type::POST);
    request->setResponseCallback(CC_CALLBACK_2(GameScene::onHttpRequestCompleted, this));

    //setHeader
    std::vector<std::string> headers;
    headers.push_back("Cookie: GAMESESSIONID=" + Global::gameSessionId);
    request->setHeaders(headers); //将sessionID加入到http请求的头部
    request->setTag("POST submit"); //设置http请求的tag

    // write the post data
    string str = "score=" + score_field->getString();
    const char* postData = str.c_str();
    request->setRequestData(postData, strlen(postData));

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

这是 submit 对应的 HttpRequest 的回调函数

```
void GameScene::onHttpRequestCompleted(HttpClient *sender, HttpResponse *response) {
    if (!response)
    {
        return;
    }
    int statusCode = response->getResponseCode();
    log("response code: %d", statusCode);
    if (!response->isSucceed())
    {
        log("response failed");
        log("error buffer: %s", response->getErrorBuffer());
        return;
    }
    std::vector<char> *buffer = response->getResponseData();
    for (unsigned int i = 0; i < buffer->size(); i++) {
        printf("%c", (*buffer)[i]);
    }
    printf("\n");
}
```

这是 rank_button 的点击事件函数

```
void GameScene::rank() {
    HttpRequest* request = new HttpRequest();
    request->setUrl("http://localhost:8080/rank?top=5");
    request->setRequestType(HttpRequest::Type::GET);
    request->setResponseCallback(CC_CALLBACK_2(GameScene::onHttpRequestCompletedrank, this));

    //setHeader
    std::vector<std::string> headers;
    headers.push_back("Cookie: GAMESESSIONID=" + Global::gameSessionId);
    request->setHeaders(headers); //将sessionID加入到http请求的头部
    request->setTag("GET rank"); //设置http请求的tag

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

这是 rank 对应的 HttpRequest 的回调函数，这里对从服务器传回来的数据进行了字符串处理

```
void GameScene::onHttpRequestCompletedrank(HttpClient *sender, HttpResponse *response) {
    std::vector<char> *databuffer = response->getResponseData();
    string rank = Global::toString(databuffer);
    rapidjson::Document d;
    d.Parse<0>(rank.c_str());
    if (d.HasParseError()) {
        rank_field->setText("Error\n");
    }
    if (d.IsObject() && d.HasMember("info")) {
        string temp = d["info"].GetString();
        string member = "";
        string allmember = "";
        for (int i = 1; i < temp.length(); i++) {
            if (temp[i] != '|') {
                member += temp[i];
            }
            else {
                allmember += member;
                allmember += '\n';
                member = "";
            }
        }
        rank_field->setText(allmember);
    }
}
```

- 完成 Bonus, “将本次作业和以往作业结合, 组成一个游戏”

三.实验结果截图

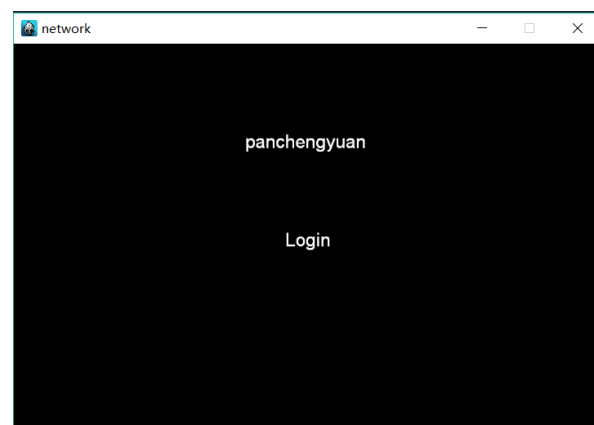
1. 下面是完成基本要求的截图

因为我已经用 TA 的 demo 输入了一些数据, 所以“info”的值为当前分数的最大值“500”

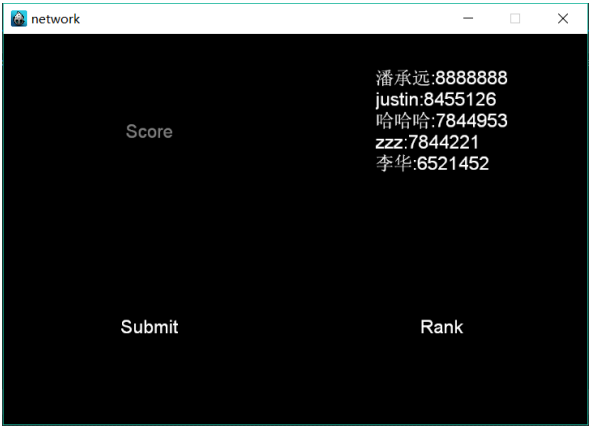


2.完成“冲击高分”部分, “实现使用用户名登录的功能”、“实现提交分数的功能”、“实现查询最好的 n 位成绩的功能”

输入“panchengyuan”



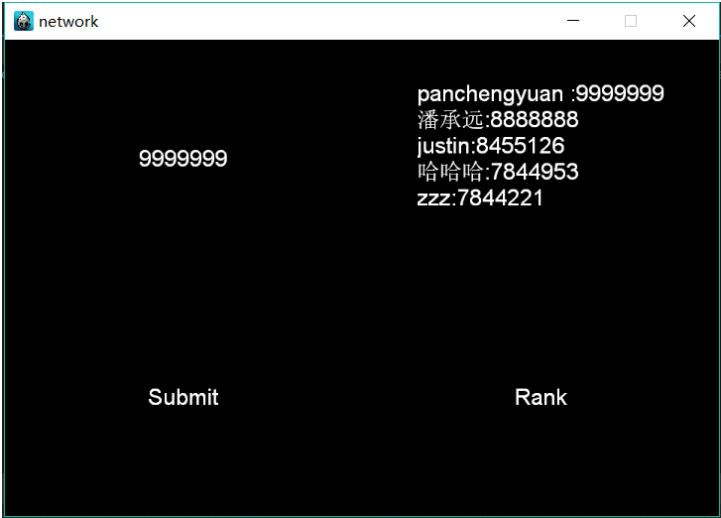
进入到游戏界面，首先点击 Rank，显示当前已经存在的分数最高的 5 个人



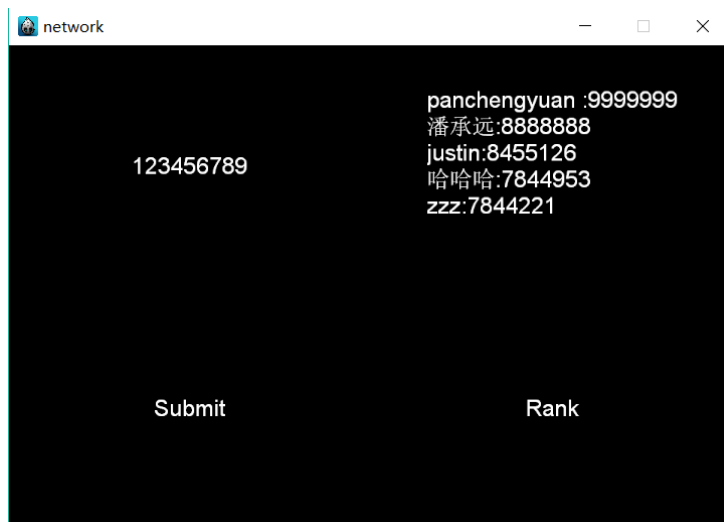
将用户名为“panchengyuan”的用户的分数设置为 9999999



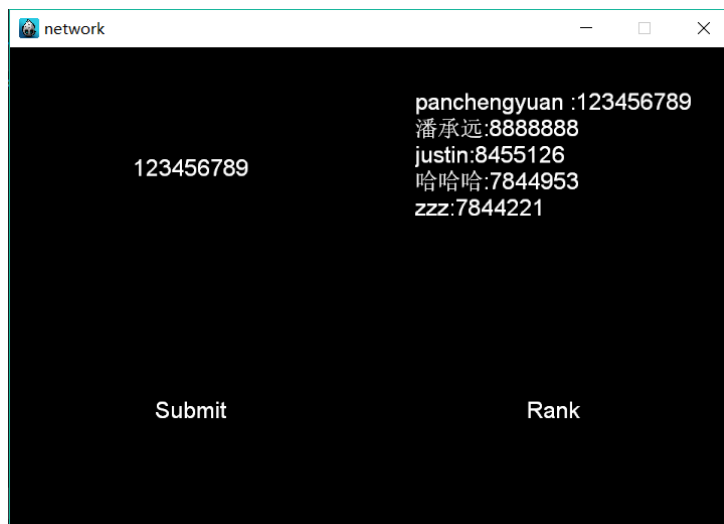
点击“Submit”，之后点击“Rank”，可以看见“panchengyuan”排到了第一位



再次输入“123456789”



点击“Submit”,接着点击“Rank”,可以看见“panchengyuan”的分数更新为最高的分数,此时如果输入比较小的分数,服务器记录的用户分数是不会改变的,因为服务器端的代码已经实现了记录用户最高分数的功能。



3. 下面是完成加分项 Bonus 的截图

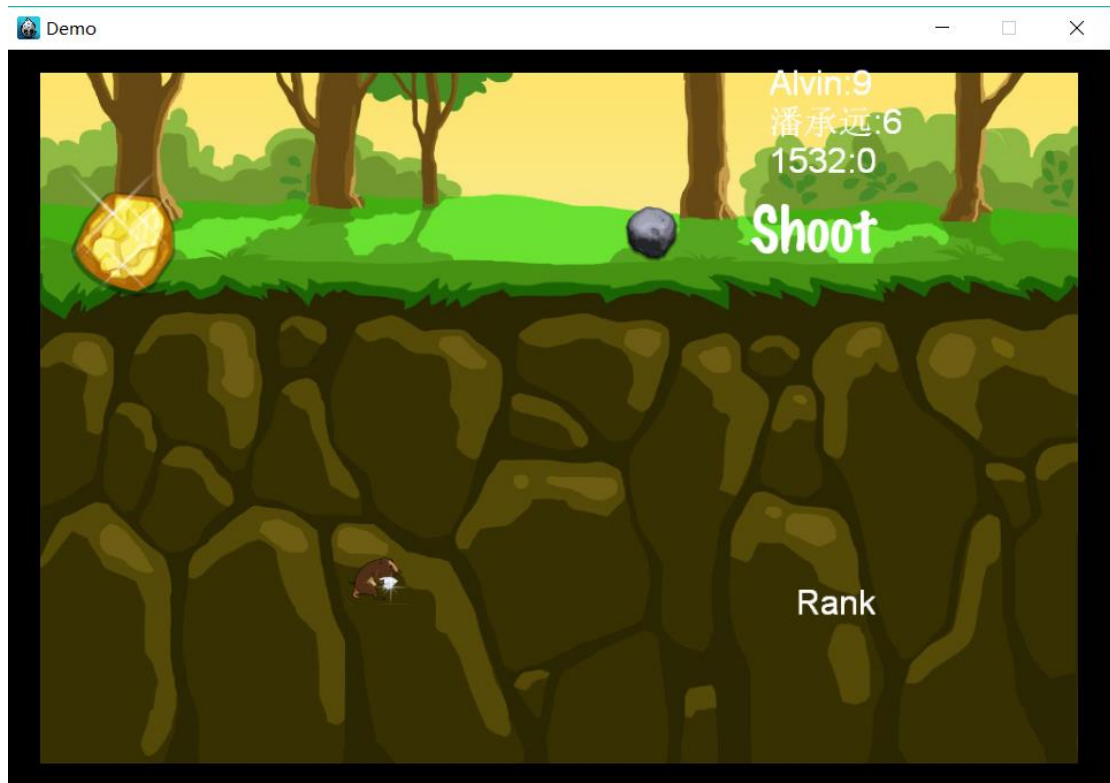
初始界面,中间的“player name”代表输入用户名的地方



输入"Alvin"



点击屏幕 9 次，代表老鼠吃了 9 块奶酪，此时 Alvin 排第一



下面是 Bonus 的 login 代码:

```
79
80 void MenuScene::startMenuCallback(cocos2d::Ref* pSender) {
81     HttpRequest* request = new HttpRequest();
82     request->setUrl("http://localhost:8080/login");
83     request->setRequestType(HttpRequest::Type::POST);
84     request->setResponseCallback(CC_CALLBACK_2(MenuScene::onHttpRequestCompleted, this));
85
86     string str = "username=" + textField->getString();
87     const char* postData = str.c_str();
88     request->setRequestData(postData, strlen(postData));
89     request->setTag("POST login");
90
91     cocos2d::network::HttpClient::getInstance()->send(request);
92     request->release();
93     auto scene = GameScene::createScene();
94     Director::getInstance()->replaceScene(TransitionFade::create(2, scene));
95 }
96
97 void MenuScene::onHttpRequestCompleted(HttpClient *sender, HttpResponse *response) {
98     if (!response)
99     {
100         return;
101     }
102     if (!response->isSucceed())
103     {
104         log("response failed");
105         log("error buffer: %s", response->getErrorBuffer());
106         return;
107     }
108     std::vector<char> *databuffer = response->getResponseData();
109     std::vector<char> *headerbuffer = response->getResponseHeader();
110     string datatemp = Global::toString(databuffer);
111     string headertemp = Global::toString(headerbuffer);
112     Global::gameSessionId = Global::getSessionIdFromHeader(headertemp);
113 }
114
115
```

提交分数的代码和 rank 的代码在前面的基础上做微妙修改即可。

四.实验过程遇到的问题

- 遇到的最有价值的问题就是刚开始的时候不知道 Global.h 和 Global.cpp 文件的作用, 在看了两个文件的源码之后, 加上课件的知识, 才意识到这两个文件用于将登陆界面的 sessionId 保存起来, 当我们在游戏界面的时候可以通过访问 Global.h 的静态变量 gameSessionId 来知道登陆界面的 sessionId, 这样就知道当前哪个用户在访问服务器, 便于提交分数。

下面是我将 sessionId 保存到 gameSessionId 的代码实现

```
std::vector<char> *databuffer = response->getResponseData();
std::vector<char> *headerbuffer = response->getResponseHeader();
string datatemp = Global::toString(databuffer);
string headertemp = Global::toString(headerbuffer);
Global::gameSessionId = Global::getSessionIdFromHeader(headertemp);
```

- 其它遇到的问题都是一些不该错的问题, 比如写好了这样的代码

```
submit_button->addClickEventListener(CC_CALLBACK_0(GameScene::submit, this));
```

这是为了给 submit_button 添加点击事件。而 rank_button 也需要添加点击事件, 于是直接将代码复制粘贴, 忘记了将"submit_button"修改为"rank_button", 这样就导致每次点击"Submit"按钮, 就直

接显示 rank 的结果。

五.思考与总结

本次实验让自己了解了网络的一些基本概念，也学会用代码来实现客户端与服务器之间的通讯连接。可能自己以前对网络这部分知识没有深入地接触，所以在完成作业的过程中遇到了很多疑惑。每做一次实验就能学到新的知识，这次的作业也收获比较多。

提一个小小的建议，建议老师在给学弟学妹上这门课的时候，可以适当将重点放在作业的代码讲解上面。我个人感觉 pdf 给的知识有些不够具体，文字性的东西太多不易理解，其实如果理解了代码，文字性的知识可能理解起来更快速。