

现代操作系统应用开发实验报告

学号： 15331418

班级： 晚上班

姓名： 郑柏川 实验名称：Cocos2d-x 网络和常用算法

一. 参考资料

ppt

二. 实验步骤

登录部分：

```
void LoginBtnClick(Ref* pSender);  
void LoginCallback(HttpClient* sender, HttpResponse* res);
```

主要是实现这两个函数，但后面因为要实现自动登录，所以又加了一个

```
void Login(string u);
```

```
void LoginScene::LoginBtnClick(Ref * r) {  
    Login(textField->getString());  
}
```

```
void LoginScene::Login(string u)  
{  
    if (textField->getString() != "") {  
        string data = "username=" + textField->getString();  
        HttpRequest* req = new HttpRequest();  
        req->setRequestType(HttpRequest::Type::POST);  
        req->setUrl("localhost:8080/login");  
        req->setRequestData(data.c_str(), data.size());  
        req->setResponseCallback(CC_CALLBACK_2(LoginScene::LoginCallback, this));  
        HttpClient::getInstance()->send(req);  
        req->release();  
    }  
}
```

发送请求，传参数

自动登录实现：

```
if (res->isSucceed()) {
    string resHeader(res->getResponseHeader()->begin(), res->getResponseHeader()->end());
    Global::gameSessionId = Global::getSessionIdFromHeader(resHeader);
    database->setStringForKey("username", textField->getString());
    CCDirector::sharedDirector()->replaceScene(GameScene::createScene());
}
```

```
if (database->getStringForKey("username") != "") {
    textField->setString(database->getStringForKey("username"));
    LoginBtnClick(nullptr);
}
```

把用户名存起来，下次直接去本地找，然后用用户名登录

GameScene 部分:

```
bool GameScene::SubmitBtnClick(Ref * test);
void GameScene::SubmitBtnCallback(HttpClient * sender, HttpResponse * response);
void GameScene::RankBtnCallback(HttpClient * sender, HttpResponse * response);
bool GameScene::RankBtnClick(Ref * test);
```

```
bool GameScene::SubmitBtnClick(Ref * r) {
    HttpRequest* request = new HttpRequest();
    request->setUrl("http://localhost:8080/submit");
    request->setRequestType(HttpRequest::Type::POST);
    request->setResponseCallback(CC_CALLBACK_2(GameScene::SubmitBtnCallback, this));
    string postData = "score=" + score_field->getString();
    request->setRequestData(postData.c_str(), postData.size());

    vector<string> header;
    header.push_back("Cookie: GAMESESSIONID=" + Global::gameSessionId);
    request->setHeaders(header);

    HttpClient::getInstance()->send(request);
    request->release();
    return true;
}
```

也是发请求传参数的套路

```

void GameScene::SubmitBtnCallback(HttpClient * sender, HttpResponse * response)
{
    std::vector<char> * buffer = response->getResponseData();
    string str;
    rapidjson::Document d;
    str = Global::toString(buffer);
    d.Parse<0>(str.c_str());
    if (d.IsObject() && d.HasMember("info")) {
        Global::score = atol(d["info"].GetString());
        char a[100];
        sprintf(a, "%ld", Global::score);
        score_field->setString(a);
    }
}

```

Submit 之后，将返回的最高分数替换掉 score 的值

```

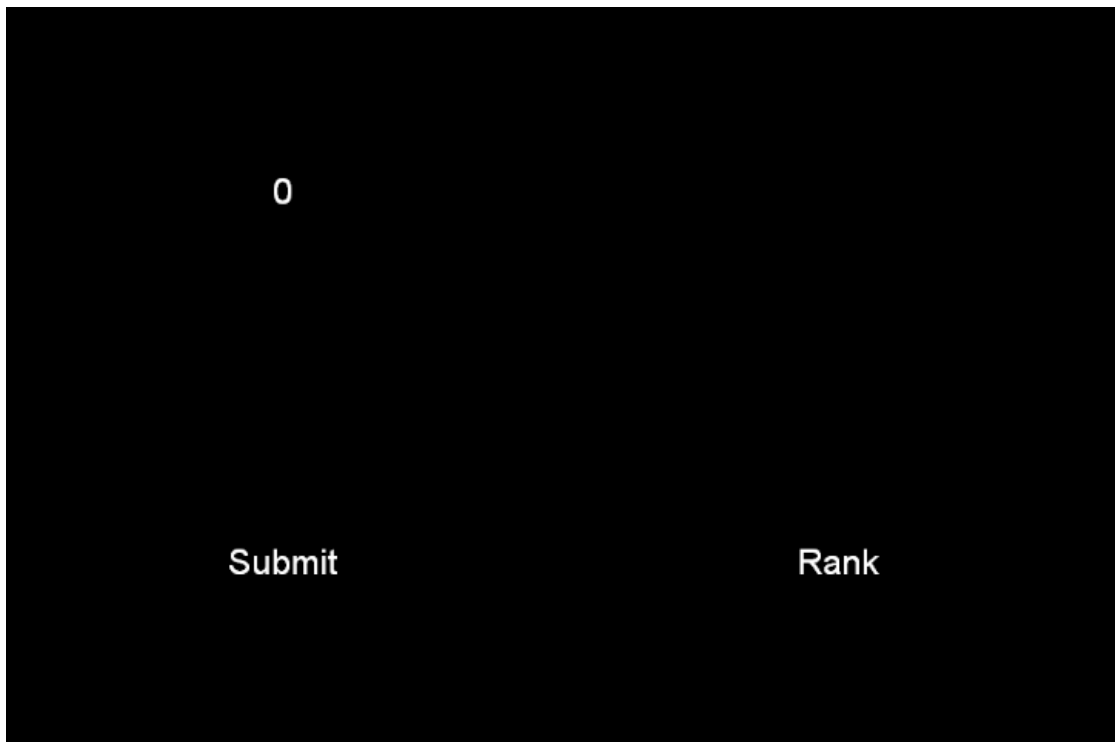
vector<char> *_header = response->getResponseHeader();
vector<char> *_body = response->getResponseData();
string header = Global::toString(_header);
string body = Global::toString(_body);
Document d;
d.Parse<0>(body.c_str());

if (d["result"].GetBool()) {
    string result = d["info"].GetString();
    for (int i = 0; i < result.length(); i++)
        if (result[i] == '|')
            result[i] = '\n';
    rank_field->setText(result);
}

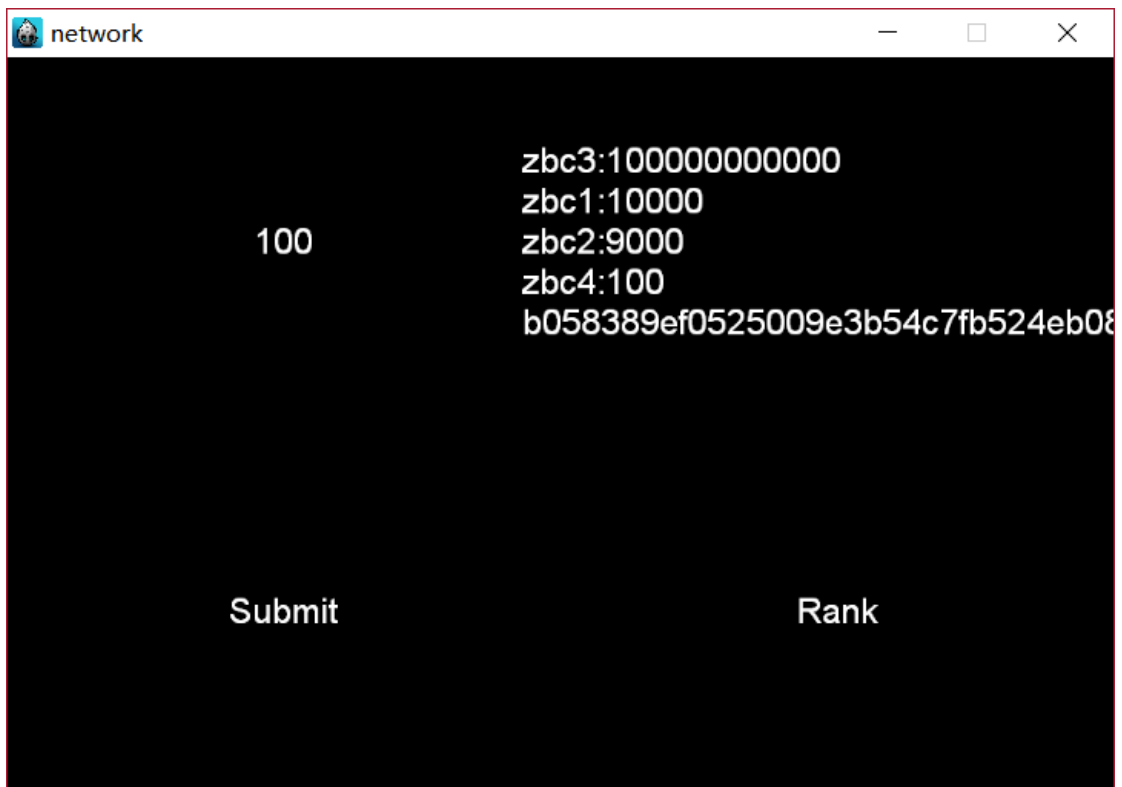
```

Rank 的话则是将传回来的解析出来

三．实验结果截图



（我是 zbc4）



（不要在意那个 id，是我在一开始写的时候写错了不小心把 sessionid 当 username 存进去了，不知道怎么清除）

四．思考与总结

本次作业没有遇到什么问题，主要是期中项目已经写过类似的网络编程，对 json 的解析，发送请求什么的都比较熟。