

《现代操作系统应用开发》HW15 实验报告

姓名：羊伊 学号：15331349

一、参考资料：

homework15.pdf

网络与常用算法.pdf

二、实验步骤：

1、实现“使用用户名登录”的功能

(1) 添加一个 textfield 作为输入用户名的地方，并添加监听。

```
tf = TextFieldTTF::textFieldWithPlaceholder("username", "arial.ttf", 35);
tf->setPosition(Vec2(login->getPositionX(), login->getPositionY()+50));

addChild(tf, 5);

auto * lis = EventListenerTouchOneByOne::create();
    event listener (onebyone)

lis->onTouchBegan = [&](Touch *t, Event *e)
{
    if (tf->getBoundingBox().containsPoint(t->getLocation()))
    {
        tf->attachWithIME();
    }
    else
    {
        tf->detachWithIME();
    }
    return false;
};

Director::getInstance()->getEventDispatcher()->
addEventListenerWithSceneGraphPriority(lis, tf);
```

(2) 添加登陆的按钮

```
auto login = MenuItemLabel::create(Label::createWithTTF(ttfConfig, "LOGIN"), CC_CALLBACK_1(HelloWorld::
loginCallback, this));
```

(3) 在回调函数中创建网络请求：

请求 URL：<http://localhost:8080/login>

Method: POST

参数：username=+输入框中输入的 name

当 isLogin 为 false 的时候怪物不会出现，所以在回调函数中将 isLogin 设为 true，怪物出现，血条设为满，当前成绩为 0，游戏开始。

```
void HelloWorld::loginCallback(Ref* pSender) {
    HttpRequest* request = new HttpRequest();
    request->setRequestType(HttpRequest::Type::POST);
    request->setUrl("http://localhost:8080/login");
    auto temp_1 = "username="+tf->cocos2d::Label::getString();
    const char* postData = temp_1.c_str();
    request->setRequestData(postData, strlen(postData));
    request->setResponseCallback(CC_CALLBACK_2(HelloWorld::onHttpComplete_login, this));
    request->setTag("LOGIN");
    cocos2d::network::HttpClient::getInstance()->send(request);
    request->release();
    isLogin = true;
    killed = 0;
    String* temp = String::createWithFormat("%d", killed);
    time->setString(temp->_string);
    database->setIntegerForKey("value", killed);
    percent = 100;
    pT->runAction(CCProgressTo::create(0.1, percent));
}
```

(4) 处理网络回调函数

将返回的 Header 中的 Gamesessionid 保存到 Gamesessionid 中。将返回的 data 保存到 string 中，然后用 rapidjson 解析它，获取目前用户的最好成绩，并将其设为 myscore 的内容。

```
void HelloWorld::onHttpComplete_login(HttpClient *sender, HttpResponse *response) {
    if (!response) {
        return;
    }
    if (!response->isSucceed()) {
        return;
    }
    std::vector<char> *header = response->getResponseHeader();
    std::string _head;
    for (unsigned int i = 0; i < header->size(); i++) {
        _head += (*header)[i];
    }
    Gamesessionid = "";
    for(int i = _head.find("GAME"); _head[i] != '\r'; i++) {
        Gamesessionid += _head[i];
    }
    Gamesessionid = "Cookie: "+Gamesessionid;
    printf("%s\n", Gamesessionid.c_str());
    std::vector<char> *data = response->getResponseData();
    std::string _data;
    for (unsigned int i = 0; i < data->size(); i++) {
        _data += (*data)[i];
    }
    rapidjson::Document d;
    d.Parse<0>(_data.c_str());
    if (d.IsObject() && d.HasMember("info")) {
        highscore = d["info"].GetString();
        myscore->setString(highscore._string);
    }
}
```

2、实现“提交分数”的功能

URL : <http://localhost:8080/submit>

Method: POST

参数 : score=+当前成绩 , 即为 killed。

在这里要在请求头里加上 GAMESESSIONID。

```
void HelloWorld::submitCallback(Ref* pSender) {
    HttpRequest* request = new HttpRequest();
    request->setRequestType(HttpRequest::Type::POST);
    request->setUrl("http://localhost:8080/submit");
    String* temp = String::createWithFormat("%d", killed);
    std::string t = "score="+temp->_string;
    const char* postData = t.c_str();
    request->setRequestData(postData, strlen(postData));
    std::vector<std::string> id;
    id.push_back(GameSessionid);
    request->setHeaders(id);
    request->setResponseCallback(CC_CALLBACK_2(HelloWorld::onHttpComplete_submit, this));
    request->setTag("SUBMIT");
    cocos2d::network::HttpClient::getInstance()->send(request);
    request->release();
}
```

3、实现 “查询最好 10 位成绩” 的功能

因为方法是 GET , 所以要把请求的内容加到 url 中。

URL : <http://localhost:8080/rank?top=10>

Method: GET

参数 : top=10

在这里要在请求头里加上 GAMESESSIONID。

```
void HelloWorld::rankCallback(Ref* pSender) {
    HttpRequest* request = new HttpRequest();
    request->setRequestType(HttpRequest::Type::GET);
    request->setUrl("http://localhost:8080/rank?top=10");
    std::vector<std::string> id;
    id.push_back(GameSessionid);
    const char* postData = "top=10";
    request->setRequestData(postData, strlen(postData));
    request->setHeaders(id);
    request->setResponseCallback(CC_CALLBACK_2(HelloWorld::onHttpComplete_rank, this));
    request->setTag("RANK");
    cocos2d::network::HttpClient::getInstance()->send(request);
    request->release();
}
```

在网络回调函数中 , 解析出 info 的内容 , 并显示到屏幕上 ,

```

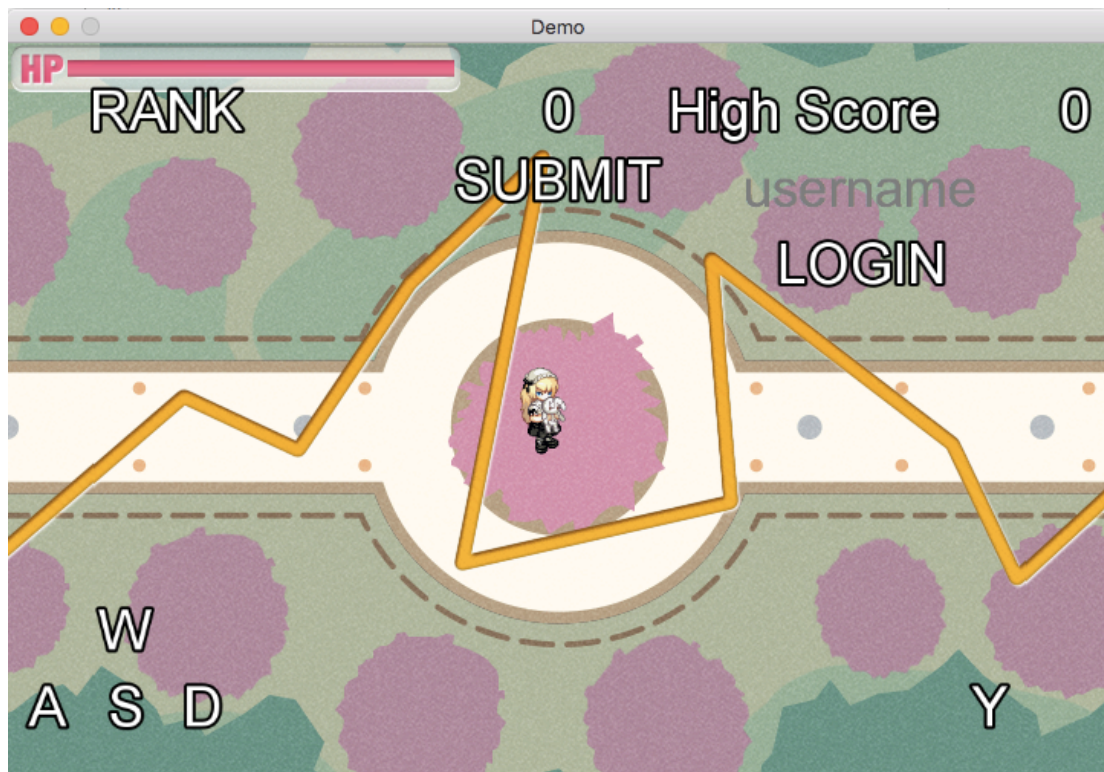
void HelloWorld::onHttpComplete_rank(HttpClient *sender, HttpResponse *response) {
    if (!response) {
        return;
    }
    if (!response->isSucceed()) {
        return;
    }
    std::vector<char> *data = response->getResponseData();
    std::string _data;
    for (unsigned int i = 0; i < data->size(); i++) {
        _data += (*data)[i];
    }
    rapidjson::Document d;
    d.Parse<0>(_data.c_str());
    if (d.IsObject() && d.HasMember("info")) {
        std::string temp = d["info"].GetString();
        for (int i = 0; i < temp.length(); i++) {
            if (temp[i] == '|') {
                temp[i] = '\n';
            }
        }
    }
    if (rank == NULL) {
        rank = Label::createWithTTF(temp, "arial.ttf", 30);
        rank->setPosition(origin.x+rank->getContentSize().width, origin.y+visibleSize.height-50-rank->
            getContentSize().height);
        addChild(rank);
    } else {
        rank->setString(temp);
    }
}
}

```

在界面呈现返回的 Header 和 Body 信息

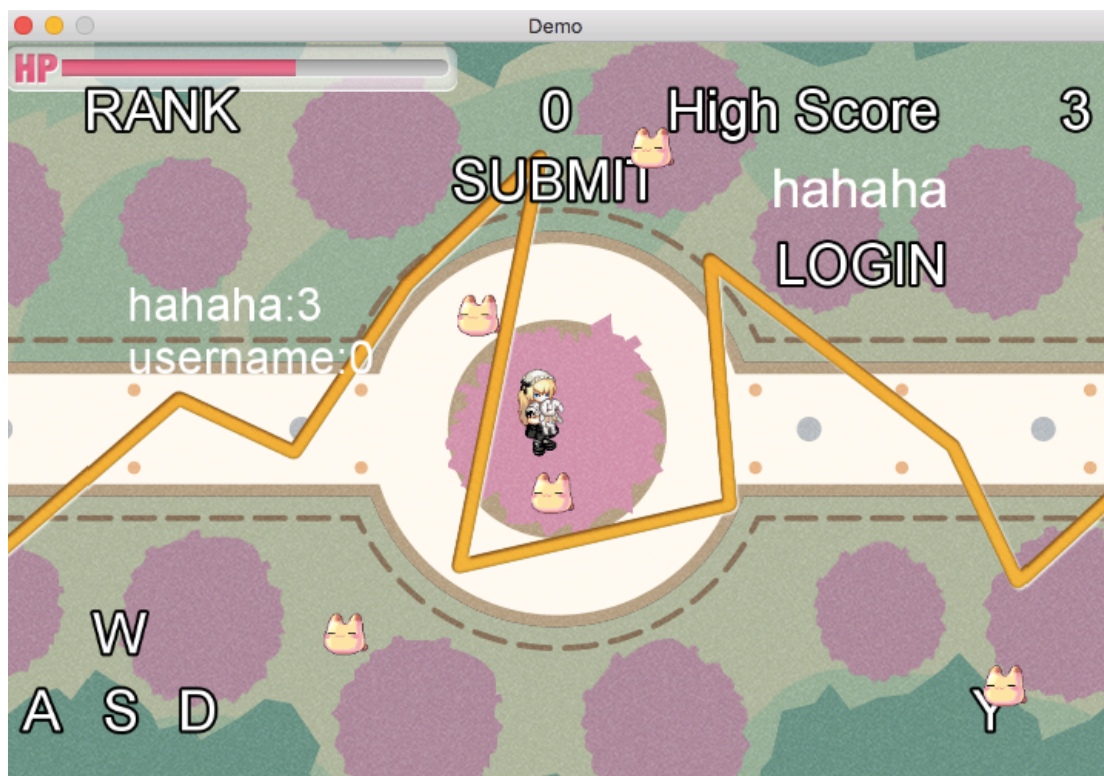
三、效果截图

最开始的状态：没有怎么搞 UI。。

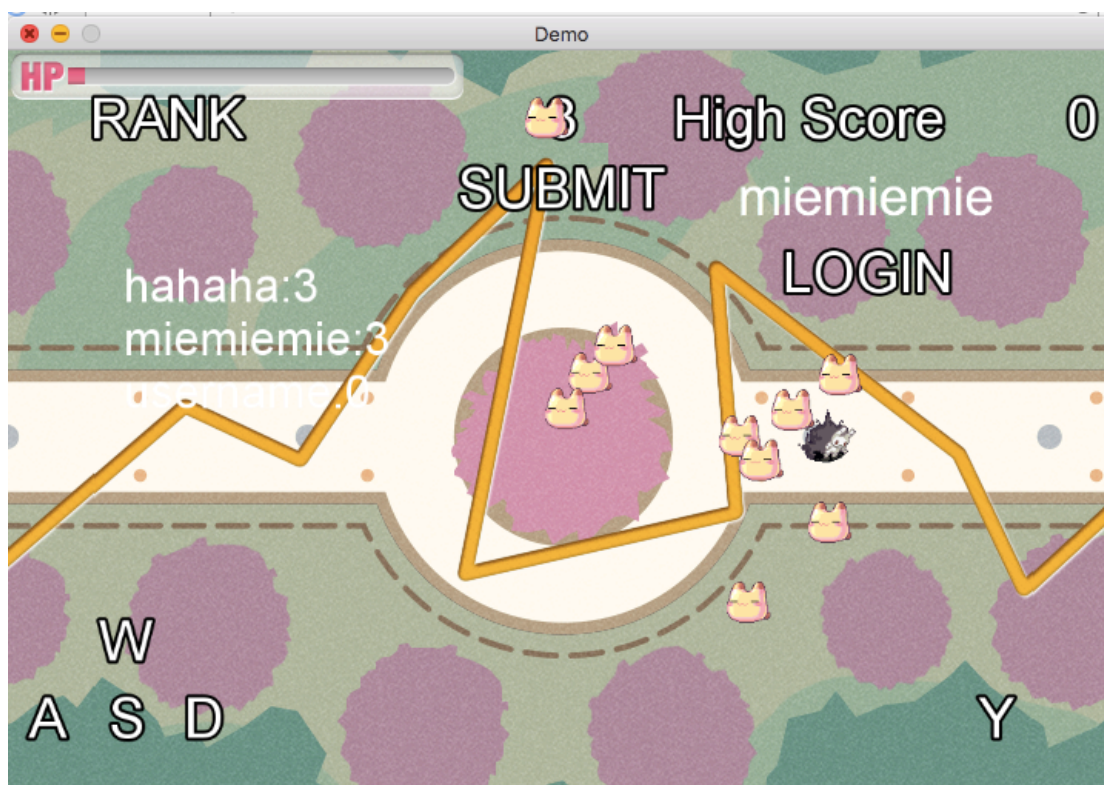


点击 RANK 显示排名。

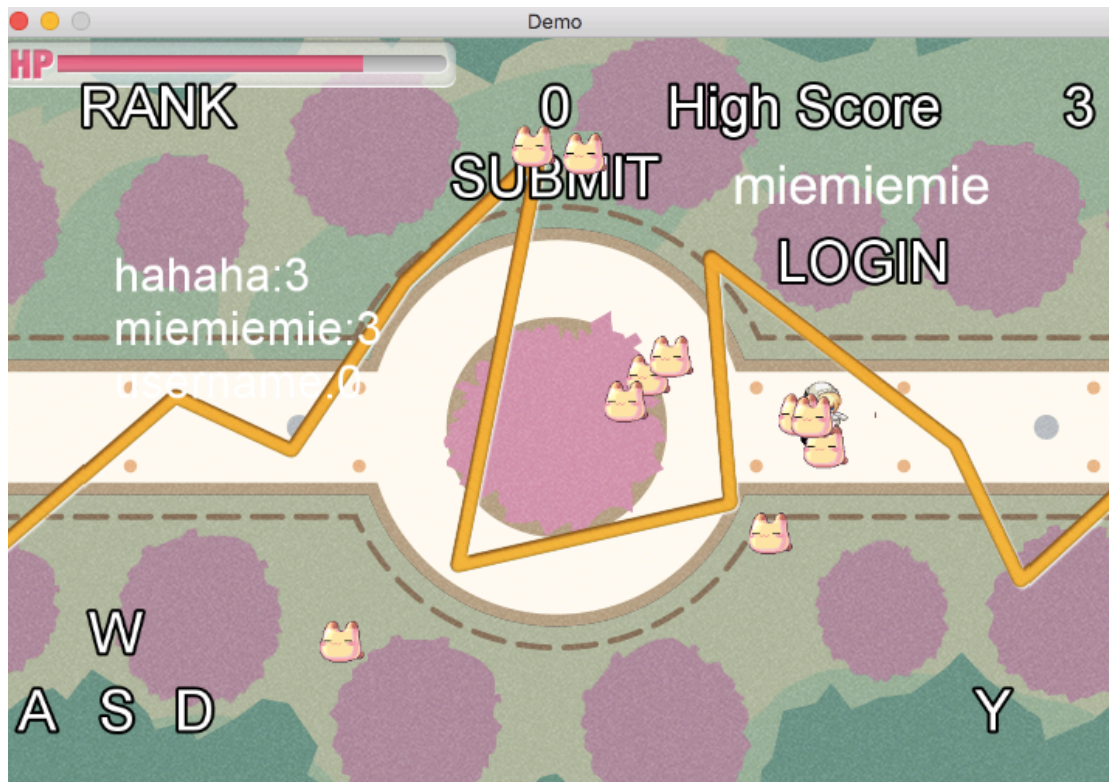
登录后，HIGHSCORE 显示最高分。



提交分数后，点击 RANK 显示在 Rank 中。



再次登录相同的账号，最高分会显示在 HIGHSCORE 上。



四、收获与感受

因为没有一个人搞 UI 的所以 UI 做的就挺糙的，有眼睇。

这次主要是做网络请求，只要按要求设置好请求的参数 Headers 和 Data 之类的，然后在返回的 Headers 和 Data 里面获取自己想要的数据就可以了。话说 rapidjson 还是挺好用的。

其他要注意的就是 GET 和 POST 的区别，GET 把请求的内容直接放在 url 中了，一开始没注意到这点掉坑里了一会。