Cpp-finalproject

This is a project made by **ThewildGangOfFour**

It's a cocos2d-x game imitating Crazy Arcade,and also a final project of TongjiUniversity School of Software engineering.

1.项目背景

2016-2017学年大一下学期面向对象编程课期末项目：使用C++结合游戏引擎制作一款游戏，本组选题为泡泡堂，选用引擎为Cocos2d-x。

2.项目分工

The list of the game's group(In no particular order):

学号	姓名	分工
1652741	杨丁豪	人物移动，碰撞检测，地图制作，动画效果,地图角色、选择，联机功能，单机模式，胜负判断，音效，UI设计，文档写作，PPT制作
1652742	贺鹏程	泡泡的放置、爆炸、地图对应变化，人物对炸弹检测及相应动画，PPT制作
1652738	赵洪城	人物属性，道具产生，拾取，使用，骑宠制作，PPT制作
1652739	李亚康	网络模块，联机功能，数据传输，PPT制作

如果具体任务量较大可以协同一起做，其他附加功能暂时没有分配，等以上功能完成后看情况决定

要求

3.功能完成情况

泡泡堂

得分权值：1.2

团队人数限制：<= 4 人

描述：实现一个类似于泡泡堂的游戏。

需要达成的基础功能：

- 支持地图绘制、人物绘制、水泡效果绘制等（√）
- 支持鼠标和键盘操作交互（√）
- 支持障碍物（√）
- 支持泡泡的放置与爆炸（√）
- 支持三种基本增强型道具（鞋子，泡泡，药水）（√）
- 实现服务端：支持局域网联机对战（自由对抗模式），且支持多人在同一个地图游戏（√）
- 支持动画效果（√）

基础功能以外的可选功能（欢迎自创）：

- 支持多个角色（角色造型和属性不同）（√）
- 支持聊天（组队聊天和所有人聊天）
- 支持房间列表（√）
- 支持 >= 2 张地图（√）
- 支持 >= 2 种游戏模式（如团队模式、大乱斗、抢包山、刺猬模式等）（√）
- 支持使用道具（如香蕉皮，飞镖）
- 支持骑宠（如乌龟，猫头鹰，飞碟，恶魔等）（√）

自行添加的功能：

- 新的道具：大力丸、金币、银币、铜币
- win32平台安装包的实现
- 音量调控（win32平台需要客户手动修改cocos引擎相关文件）
- 游戏计时（+180s）

3+.关于我们的联机模式（重点）

没钱买第三方服务器丶(≧□≦)ノ，没抢到学生服务器，所以选择了直接在本地建立一个局域网服务器的方式，使用方法：

首先确保要联机的小伙伴们同处一个局域网内（直接开热点、用无线网卡给基友开Wifi、Hamachi、或者游戏对战平台直接虚拟局域网），进入大厅界面，点击“开设房间”，自动运行服务端程序，在把自己的IP发送给要联机的小伙伴，点击“快速加入”输入对应IP，就可以进入对应房间，不同玩家创建的不同服务器使用IP进行区分，就实现了房间选择，进入游戏后，两位玩家需选择相同地图不同角色（会给出提示），再点击确定进入游戏。

联机模式可以进行本地测试，支持同一PC打开多个客户端，开启一个服务端，使用127.0.0.1的本地IP即可实现联机测试啦ヾ(￣▽￣)Bye~Bye~

4.算法描述与核心代码

(1)UI设计

使用Cocos引擎内置的Layer，Scene，Image,Text,CheckBox，Layout，Button等组件进行设计

```
auto rootNode = CSLoader::createNode("LoginScene/LoginScene.csb");//获取csb
Layout* background = (Layout*)rootNode->getChildByName("background");//获取csb上的层
Button* btnMenu = (Button*)Helper::seekWidgetByName(background, "menubutton");//获取层上的button
```

(2)游戏界面

使用Cocos的Sprite类，在PlayScene中加入Sprite作为角色，地图使用Tiled制作的TMX瓦片地图

```
//load the plist file and init the role
    cache = SpriteFrameCache::getInstance();
    if (role_tag == 1 && test_model == false)
        isHost = true;
    role1.roleInit(objects, cache, role_tag,false);
    role2.roleInit(objects, cache, (role_tag == 1)?2:1,true);
    gameMap->addChild(role1.role, 3);
    gameMap->addChild(role2.role, 3);
    m_Roles[0] = &role1;
    m_Roles[1] = &role2;
```

(3)人物移动与碰撞检测

人物移动采用在PlaySCene中设置cocos的scheduleUpdate调度器每一帧调用一次检测，同时开启键盘监听，如果有对应的方向键按下，就设置MoveTo动作并开始播放对应的序列帧动画

```
//调度器
void MapOfGame::update(float delta) {
    Node::update(delta);
    role1.loadPositon();
    auto upArrow = EventKeyboard::KeyCode::KEY_UP_ARROW,
        downArrow = EventKeyboard::KeyCode::KEY_DOWN_ARROW,
        leftArrow = EventKeyboard::KeyCode::KEY_LEFT_ARROW,
        rightArrow = EventKeyboard::KeyCode::KEY_RIGHT_ARROW;
    if (isKeyPressed(upArrow)) {
        keyPressedMovement(upArrow);
    }
    else if (isKeyPressed(downArrow)) {
        keyPressedMovement(downArrow);
    }
    else if (isKeyPressed(leftArrow)) {
        keyPressedMovement(leftArrow);
    }
    else if (isKeyPressed(rightArrow)) {
        keyPressedMovement(rightArrow);
    }
}

........
  //对应的移动函数
  void MapOfGame::keyPressedMovement(EventKeyboard::KeyCode keyCode) {
    CCPoint moveByPosition;
    RoleDirection tag;
    //you can set move speed here
    switch (keyCode) {
    case EventKeyboard::KeyCode::KEY_UP_ARROW:
        moveByPosition = ccp(0, role1.getSpeed());
        break;
    case EventKeyboard::KeyCode::KEY_DOWN_ARROW:
        moveByPosition = ccp(0, -role1.getSpeed());
        break;
    case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
        moveByPosition = ccp(-role1.getSpeed(), 0);
        break;
    case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
        moveByPosition = ccp(role1.getSpeed(), 0);
        break;
    default:
        moveByPosition = ccp(0, 0);
        break;
    }
  //创建MoveBy对象并执行移动的动作
    auto move = CCMoveBy::create(0.01f, moveByPosition);
    role1.role->runAction(move);
}
//这些是在init里面写的
    //add keyboard listener
    auto listener = EventListenerKeyboard::create();

    //call responding animation when realted key is pressed
    listener->onKeyPressed = [=](EventKeyboard::KeyCode keyCode, Event *event) {
        keys[keyCode] = true;
        switch (keyCode){
        case EventKeyboard::KeyCode::KEY_UP_ARROW:
            keyPressedAnimation(keyCode);
            break;
        case EventKeyboard::KeyCode::KEY_DOWN_ARROW:
            keyPressedAnimation(keyCode);
            break;
        case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
            keyPressedAnimation(keyCode);
            break;
        case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
            keyPressedAnimation(keyCode);
            break;
        default:
            break;
    }
};


.......
  //对应的动画函数，walkAnimation是之前弄的四个方向序列帧动画数组
  void MapOfGame::keyPressedAnimation(EventKeyboard::KeyCode keyCode) {
    RoleDirection tag;
    //you can set move speed here
    switch (keyCode) {
    case EventKeyboard::KeyCode::KEY_UP_ARROW:
        tag = kUp;
        break;
    case EventKeyboard::KeyCode::KEY_DOWN_ARROW:
        tag = kDown;
        break;
    case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
        tag = kLeft;
        break;
    case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
        tag = kRight;
        break;
    default:
        break;
    }
    animations[tag] = RepeatForever::create(CCAnimate::create(walkAnimations[tag]));
    //animations[tag] = CCAnimate::create(walkAnimations[tag]);
    role1.role->runAction(animations[tag]);
}
```

碰撞检测是通过在TMX地图上设置障碍物图层，用对应代码获取图层结合Sprite位置判断来实现的，调用MoveTo函数前，会先判断对应的TargetPosition是否有障碍物再返回能否移动，如果有障碍物则会计算可以移动的最小距离并设为新的TargetPosition，再调用对应的帧图像重置给Sprite，以实现“面壁”的效果

```
this->schedule(schedule_selector(MapOfGame::update), 0.05f);
//碰撞检测部分的代码，和之前贴的移动位置代码是一个函数
    //collision check
    CCPoint targetPosition = ccpAdd(role1.role->getPosition(), moveByPosition);
    if (checkCollision(targetPosition, tag) == kWall) {
        setFaceDirection(tag);
        return;
    }
    auto move = CCMoveBy::create(0.01f, moveByPosition);
    role1.role->runAction(move);

/****************File in CollisionCheck.cpp*****************/
#include "PlayScene.h"
#include"Player.h"
#include "CoordTransfer.h"

//collision check according to the role's position
CollisionType checkCollision(cocos2d::CCPoint rolePosition, cocos2d::CCPoint targetPosition,
RoleDirection direction) {
    CCPoint searchRange = ccp(0, 0);
    //set search range for four directioin
    switch (direction)
    {
    case kUp:
        searchRange = ccp(0, 5);
        break;
    case kDown:
        searchRange = ccp(0, -28);
        break;
    case kLeft:
        searchRange = ccp(-20, 0);
        break;
    case kRight:
        searchRange = ccp(15, 0);
        break;
    default:
        break;
    }
    auto roleTilePosition = tilecoordForPosition(rolePosition);
    targetPosition += searchRange;
    //transfer the coord
    CCPoint tileCoord;
    //check the border of map
    if (targetPosition.x<0 || targetPosition.x>theMap->getMapSize().width*theMap-
>getTileSize().width
        || targetPosition.y<0 || targetPosition.y>theMap->getMapSize().height*theMap-
>getTileSize().height) {
        return kWall;
    }
    //check the obstacles
    //added bombs
    if (direction == kUp || direction == kDown) {

        for (int i = 1, j = -1; i <= 2; i++, j *= -1) {
            if(targetPosition.x <= 13.5)
                break;
            searchRange = ccp(13.5* j*i, 0);
            targetPosition += searchRange;
            tileCoord = tilecoordForPosition(targetPosition);
            if (theMap->layerNamed("architecture-real")->tileGIDAt(tileCoord)) {
                return kWall;
            }
            //bomb check
            for (int i = 0; i < 2; i++)
            {
                for (auto it :m_Roles[i]->m_Bombs)
                {
                    if (it->droppedOrNot())
                    {
                        auto bombTilePosition = tilecoordForPosition(it->bombOpenglCoord());
                        if (roleTilePosition == bombTilePosition)
                            continue;
                        else if (tileCoord == bombTilePosition)
                            return kWall;
                    }
                }
            }
        }
    }
    else if (direction == kLeft || direction == kRight) {
        for (int i = 1 ; i <= 2 ; i++) {
            if (targetPosition.y < 27.6)
                break;
            if (i == 1)
                searchRange = ccp(0, -27);
            if (i == 2)
                searchRange = ccp(0, 29);
            targetPosition += searchRange;
            tileCoord = tilecoordForPosition(targetPosition);
            if (theMap->layerNamed("architecture-real")->tileGIDAt(tileCoord)) {
                return kWall;
            }
        }
        //bomb check
        for (int i = 0; i < 2; i++)
        {
            for (auto it :m_Roles[i]->m_Bombs)
            {
                if (it->droppedOrNot())
                {
                    auto bombTilePosition = tilecoordForPosition(it->bombOpenglCoord());
                    if (roleTilePosition == bombTilePosition)
                        continue;
                    else if (tileCoord == bombTilePosition)
                        return kWall;
                }

            }
        }
    }

    return kNone;
}
```

(4)炸弹的实现

为了实现炸弹的爆炸、计时、计数、相互引爆等功能，专门创建了一个Bomb类，并采用has-a关系在Player类中创建一个Bomb对象

```
#ifndef _BOMB_H_
#define _BOMB_H_


#include "cocos2d.h"
#include "cocostudio/CocoStudio.h"
#include "ui/CocosGUI.h"
#include <string>
USING_NS_CC;
#include <SimpleAudioEngine.h>
using namespace CocosDenshion;

class cBomb:public cocos2d::Layer
{
private:
    int m_IdleTime;
    int m_BombRange;
    float m_CurrentTime;
    bool m_Dropped;
    bool m_Exploded;
    CCPoint m_BombPosition;             //the opengl coordinate
    CCPoint m_TBombPosition;            //the tile coordinate
    cocos2d::CCTMXTiledMap* m_Map;
    cocos2d::CCSprite* m_Role1;
    enum dire {
        UP = 0,
        RIGHT,
        DOWN,
        LEFT,
    };
    Vector<Sprite*> m_AllSprites;
public:
    CCPoint points[4] = { ccp(0,-1),ccp(1,0),ccp(0,1),ccp(-1,0) };
    int m_Board[4];
    cBomb(int bombRange = 1, int idleTime = 3) :m_IdleTime(idleTime),
        m_BombRange(bombRange), m_CurrentTime(0.0),
m_Dropped(false),m_Map(nullptr),m_Role1(nullptr),
        m_Exploded(false)
    {
        cocos2d::Layer::onEnter();
    }
    void getMap(cocos2d::CCTMXTiledMap* map)
    {
        m_Map = map;
    }
    void getRole(cocos2d::CCSprite* role)
    {
        m_Role1 = role;
    }
    cocos2d::CCTMXTiledMap* returnMap()
    {
        return m_Map;

    }
    cocos2d::CCSprite* returnRole()
    {
        return m_Role1;
    }
    bool droppedOrNot()
    {
        return m_Dropped;
    }
    CCPoint bombOpenglCoord()
    {
        return m_BombPosition;
    }
    void detonate()
    {
        m_CurrentTime = 4.0f;
    }

    //increase the bomb power
    void addBombRange() { ++m_BombRange; }
    //get the bomb's opengl coordinate
    CCPoint getBombPosition();
    //get the role's Tile coordinate
    CCPoint getGrid();
    //create the animation of bomb before explosion
    Animation* creatBombAnimation();
    //drop the bomb
    void dropBomb();
    void dropBomb(int tx,int ty);
    //create the animation around the explosion center
    //divided into four directions
    Animation* creatExplodeAnimation(dire drection);
    //create the animation of explosion center
    Animation* creatCenterAnimation();
    //the explosion in one direction
    void explosion(dire direction);
    //the complete explosion process
    void explode();
    //Timer before explosion
    void idleUpdate(float dt);
    //Timer during the explosion
    void explodeUpdate(float dt);
    //remove the tile in m_Map destoied by explosion
    void removeTile(dire direction);

    bool explodedOrNot()
    {
        return m_Exploded;
    }
    CCPoint showBombPosition()
    {
        return m_BombPosition;
    }

    void setBombRange(int bombRange)
    {
        m_BombRange = bombRange;
    }
    int showBombRange()
    {
        return m_BombRange;
    }
};

#endif;
```

（4）道具的实现

道具Item也是通过创建Sprite并在map上对应位置addchild实现，在Bomb的爆炸函数处提取到moveTile的坐标，然后调用随机数函数随机产生“1-9”的随机数用于生成对应道具，而道具类中加入了坐标检测函数，但角色Sprite与道具Sprite的坐标重合后就remove Sprite并对应改变角色属性，骑宠则还要更换预先加载入缓存的动画。

```
#ifndef __ITEM_H__
#define __ITEM_H__
#include "cocos2d.h"
#define PROBABILITY 0.8

class Item : public cocos2d::Node
{
public:
    Item(int &itemNo, const cocos2d::CCPoint &TPos, cocos2d::CCTMXTiledMap* Map);
    ~Item() { remove(); }
    void remove();
    cocos2d::CCPoint getItemPosition();
private:
    int itemNo;
    cocos2d::Sprite* item;
    cocos2d::Sprite* shadow;
    cocos2d::CCTMXTiledMap* Map;
    cocos2d::CCPoint tilePos;
};
//random number generator
unsigned int randNum();

//random item at a random place
void randomItem(const cocos2d::CCPoint &itemPos, cocos2d::CCTMXTiledMap* Map);

#endif; //__ITEM_H__


#include "Item.h"
#include "Data.h"

int isItem[15][13] = { 0 };
Item* items[15][13] = { nullptr };

Item::Item(int &itemno, const cocos2d::CCPoint &TPos, cocos2d::CCTMXTiledMap* m_Map)
:tilePos(TPos), Map(m_Map),itemNo(itemno)
{
    cocos2d::Node::onEnter();

    //create item
    auto position = getItemPosition();
    auto itemName = cocos2d::String::createWithFormat("Item/item%d__.png", itemNo);
    item = cocos2d::Sprite::create(itemName->getCString());
    item->setAnchorPoint(cocos2d::Vec2(0, 0));
    item->setPosition(position);
    item->setVisible(true);
    Map->addChild(item, 2.5);

    //transport the item information
    if(isHost)
        itemInfo[static_cast<int>(TPos.y * 15 + TPos.x)] = itemNo + 48;

    //make items jump up and down
    cocos2d::JumpBy* jumpby = cocos2d::JumpBy::create(1, cocos2d::Vec2(0, 0), 3, 1);
    cocos2d::RepeatForever *repeatforever = cocos2d::RepeatForever::create(jumpby);
    item->runAction(repeatforever);

    //create shadow
    shadow = cocos2d::Sprite::create("Item/shadow__.png");
    shadow->setAnchorPoint(cocos2d::Vec2(0, 0.5));
    shadow->setPosition(position);
    item->setVisible(true);
    Map->addChild(shadow, 1);

    cocos2d::Vector<cocos2d::SpriteFrame*> shadowArry;
    auto *frameCache1 = cocos2d::SpriteFrameCache::getInstance();
    frameCache1->addSpriteFramesWithFile("Item/shadow.plist", "Item/shadow1.png");
    for (int i = 1; i < 3; ++i)
    {
        auto *frame1 = frameCache1-
>getSpriteFrameByName(cocos2d::String::createWithFormat("shadow_%d.png", i)->getCString());
        shadowArry.pushBack(frame1);
    }
    cocos2d::Animation *animation1 = cocos2d::Animation::createWithSpriteFrames(shadowArry);
    animation1->setLoops(-1);
    animation1->setDelayPerUnit(0.5f);

    auto *action1 = cocos2d::Animate::create(animation1);
    shadow->runAction(action1);

    if (itemNo > 3)
    {
        auto plistFile = cocos2d::String::createWithFormat("Item/item%d.plist", itemNo);
        auto pngFile = cocos2d::String::createWithFormat("Item/item%d.png", itemNo);

        auto *frameCache = cocos2d::SpriteFrameCache::getInstance();
        frameCache->addSpriteFramesWithFile(plistFile->getCString(), pngFile->getCString());

        cocos2d::Vector<cocos2d::SpriteFrame*> itemArry;
        for (int i = 1; i < 4; ++i)
        {
            auto *frame = frameCache-
>getSpriteFrameByName(cocos2d::String::createWithFormat("item%d_%d.png", itemNo, i)-
>getCString());
            itemArry.pushBack(frame);
        }
        cocos2d::Animation *animation = cocos2d::Animation::createWithSpriteFrames(itemArry);
        animation->setLoops(-1);
        animation->setDelayPerUnit(0.1f);

        auto *action = cocos2d::Animate::create(animation);
        item->runAction(action);
    }

    items[static_cast<int>(TPos.x)][static_cast<int>(TPos.y)] = &(*this);

    isItem[static_cast<int>(TPos.x)][static_cast<int>(TPos.y)] = itemNo;
```

```
}

/****************************************************/
void Item::remove()
{
    Map->removeChild(item);
    Map->removeChild(shadow);
}

cocos2d::CCPoint Item::getItemPosition()
{
    float x = static_cast<float>(tilePos.x) * Map->getTileSize().width;
    float y = (Map->getMapSize().height - 1 - static_cast<float>(tilePos.y)) * Map-
>getTileSize().height;
    return cocos2d::Vec2(x, y);
}

unsigned int randNum()
{
    static HCRYPTPROV hProvider = 0;
    static const DWORD dwLength = 2;
    static BYTE pbBuffer[dwLength] = {};

    DWORD result = ::CryptAcquireContextW(&hProvider, 0, 0, PROV_RSA_FULL, CRYPT_VERIFYCONTEXT |
CRYPT_SILENT);

    DWORD res = ::CryptGenRandom(hProvider, dwLength, pbBuffer);
    auto randomVal = *(unsigned int*)pbBuffer;

    ::CryptReleaseContext(hProvider, 0);

    return randomVal;
}

void randomItem(const cocos2d::CCPoint &itemPos, cocos2d::CCTMXTiledMap* Map)
{
    if (test_model == true) {
        int itemNum = randNum() % 100;

        if (itemNum > 100 * (1 - PROBABILITY))
        {
            int itemNo = 0, itemNumber = 0;
            if (itemNum < 80)
                itemNo = itemNum % 4 + 1;
            else
                itemNo = itemNum % 3 + 5;
            if (itemNo > 0) {
                if (itemNo == 1)
                    itemNumber = itemNo + randNum() % 3;
                else
                    itemNumber = itemNo + 2;
            }

            auto item = new Item(itemNumber, itemPos, Map);
        }
    }
    else {
        if (fixedItems[myMapSelect - 1][static_cast<int>(itemPos.x)][static_cast<int>(itemPos.y)])
            auto item = new Item(fixedItems[myMapSelect - 1][static_cast<int>(itemPos.x)][static_cast<int>(itemPos.y)], itemPos, Map);
    }
}
```

（5）网络模块

采用了winSocket协议，客户端与游戏文件写在同一项目之中，服务端则是单独写好后与游戏对应Button用WinExec建立联系。并没有购买第三方服务器，而是可以直接在局域网中建立本地服务器并输入IP进行联机

服务端部分代码：

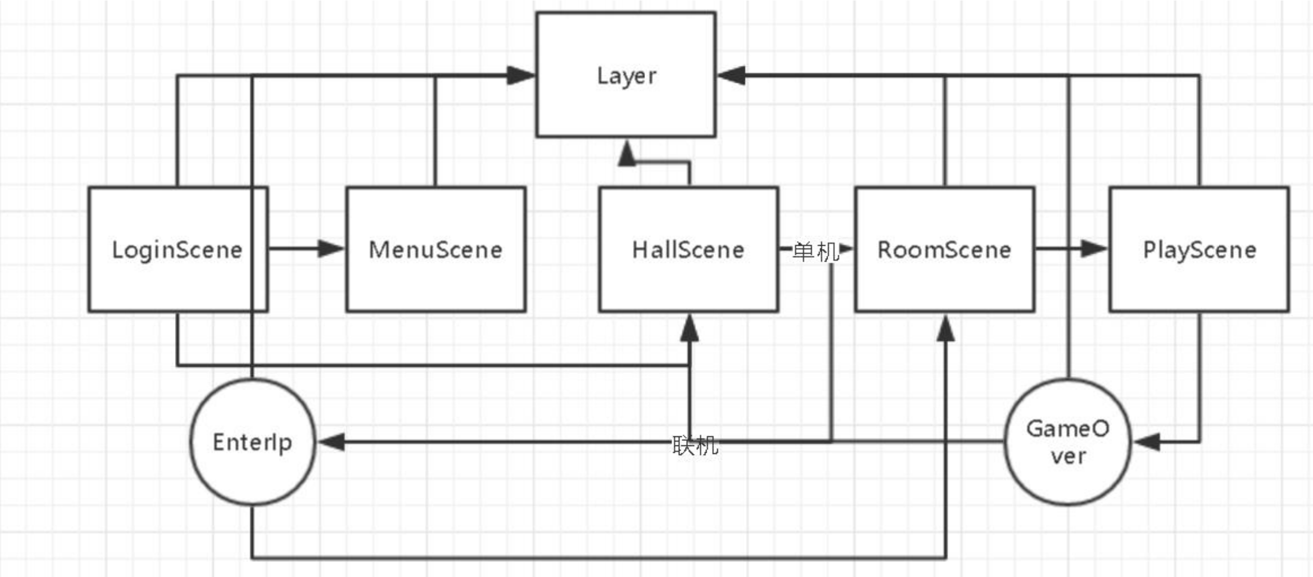
```
#include"ServerNet.h"
#include<iostream>

char receivemap[3] = { 0 };
char respose[1] = { 0 };
char recMessage1[19] = { 0 };
char recMessage2[19] = { 0 };
int false_ = 0;

int ServerNet::ServerInit(int port)
{
    int rlt = 0;

    int iErrorMsg;

    //初始化WinSock
    WSAData wsaData;
    iErrorMsg = WSAStartup(MAKEWORD(1, 1), &wsaData);

    if (iErrorMsg != NO_ERROR)
    {
        //初始化WinSock失败
        printf("server wsastartup failed with error : %d\n", iErrorMsg);

        rlt = 1;
        return rlt;
    }

    // 创建服务器端socket

    m_sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if (m_sock == INVALID_SOCKET)
        // 创建socket出现了异常
    {
        printf("server socket failed with error: %d\n", WSAGetLastError());
        rlt = 2;
        return rlt;
    }

    // 声明信息
    SOCKADDR_IN servaddr;
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = port;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);

    //绑定
    iErrorMsg = bind(m_sock, (SOCKADDR*)&servaddr, sizeof(servaddr));
    if (iErrorMsg < 0)
    {
        //绑定失败
        printf("bind failed with error : %d\n", iErrorMsg);
        rlt = 3;

        return rlt;
    }

    return rlt;
}

void ServerNet::ServerRun()
{
    // 公开连接
    listen(m_sock, 5);

    SOCKADDR_IN tcpAddr;
    int len = sizeof(sockaddr);
    SOCKET newSocket;
    int rval;

    do
    {
        // 接收信息
        newSocket = accept(m_sock, (sockaddr*)&tcpAddr, &len);

        if (newSocket == INVALID_SOCKET)
        {
            // 非可用socket
            printf("invalid socket occured.\n");
        }
        else
        {
            // 可用的新socket连接
            printf("new socket connect: %d\n", newSocket);

            do
            {
                //接收数据
                char str[19];
                memset(str, 0, sizeof(str));
                rval = recv(newSocket, str, sizeof(str), 0);
                if (rval == SOCKET_ERROR)
                    // 该异常通常发生在未closeSocket就退出时
                {
                    printf("recv socket error.\n");
                    break;
                }
                else if (rval == 0)
                    // 0表示正常退出
                    printf("socket %d connect end.\n", newSocket);
                else
                    // 显示接收到的数据
                    //std::cout << str;


                    //发送数据
                    if (str[18] == '1')

                        false_++;
                    if (str[0] == 'H') send(newSocket, "Y", 1, 0);
                    else if (str[0] == '1')
                    {
                        if (str[1] != 'M')
                        {
                            receivemap[1] = str[1];
                            respose[0] = receivemap[2];
                            if (receivemap[2] == receivemap[1]) send(newSocket, "o", 1, 0);
                            else send(newSocket, respose, 1, 0);
                        }
                        else
                        {
                            memcpy(recMessage1, str,  sizeof(str));
                            send(newSocket, recMessage2, sizeof(recMessage2), 0);
                        }
                    }
                    else if (str[0] == '2')
                    {
                        if (str[1] != 'M')
                        {
                            receivemap[2] = str[1];
                            respose[0] = receivemap[1];
                            if (receivemap[2] == receivemap[1]) send(newSocket, "o", 1, 0);
                            else send(newSocket, respose, 1, 0);
                        }
                        else
                        {

                            memcpy(recMessage2, str,  sizeof(str));
                            send(newSocket, recMessage1, sizeof(recMessage1), 0);
                        }
                    }
            } while (rval != 0);
            if (false_>1)
                break;


            // 关于接收的socket
            closesocket(newSocket);
        }
    } while (1);

    // 关闭自身socket
    closesocket(m_sock);
}
```

客户端部分代码

```
#include<iostream>
#include<string>
#include<stdio.h>
#include"ClientNet.h"
#include"client.h"
#include "../GamePlay/Data.h"

#pragma comment(lib, "Ws2_32.lib")

using namespace std;

 //全局

void Client()
{
    int rlt = 0;
    string msg;

    if (rlt == 0)
    {
        rlt = client.ClientConnect(8888, ip.c_str());

        msg = "H";
        client.ClientSend(msg.c_str(), msg.length());
        client.ClientClose();
    }
}

void choose()
{
    int rlt = 0;
    rlt = client.ClientConnect(8888, ip.c_str());
    char choose[2] = { 0 };
    choose[0] = myRoleSelect + 48;
    choose[1] = myMapSelect + 48;
    rlt = client.ClientSend(choose, 2);
    client.ClientClose();
}

void message()
{
    memset(mymessage, 0, sizeof(mymessage));
    memset(othermessage, 0, sizeof(othermessage));
    int rlt = 0;
    rlt = client.ClientConnect(8888, ip.c_str());

    mymessage[0] = myRoleSelect + 48;
    mymessage[1] = 'M';
    mymessage[2] = myPlayerInformation.isUpPressed + 48;
    mymessage[3] = myPlayerInformation.isDownPressed + 48;
    mymessage[4] = myPlayerInformation.isLeftPressed + 48;
    mymessage[5] = myPlayerInformation.isRightPressed + 48;

    mymessage[6] = myPlayerInformation.isSpacePressed + 48;
    //add
    int x_hun = myPlayerInformation.role_x / 100;
    int x_ten = (myPlayerInformation.role_x - x_hun * 100) / 10;
    int x_ge = myPlayerInformation.role_x - x_ten * 10 - x_hun * 100;
    mymessage[7] = x_hun + 48;
    mymessage[8] =x_ten + 48;
    mymessage[9] = x_ge + 48;
    int y_hun = myPlayerInformation.role_y / 100;
    int y_ten = (myPlayerInformation.role_y - y_hun * 100) / 10;
    int y_ge = myPlayerInformation.role_y - y_ten * 10 - y_hun * 100;
    mymessage[10] = y_hun + 48;
    mymessage[11] = y_ten + 48;
    mymessage[12] = y_ge + 48;
    mymessage[13] = myPlayerInformation.bomb_x + 48;
    mymessage[14] = myPlayerInformation.bomb_y + 48;
    int t_hun = gameTime / 100;
    int t_ten = (gameTime - t_hun * 100) / 10;
    int t_ge = gameTime - t_hun * 100 - t_ten * 10;
    mymessage[15] = t_hun + 48;
    mymessage[16] = t_ten + 48;
    mymessage[17] = t_ge + 48;
    mymessage[18] = closeServer + 48;


    rlt = client.ClientSend(mymessage,sizeof(mymessage));

    client.ClientClose();
}
```

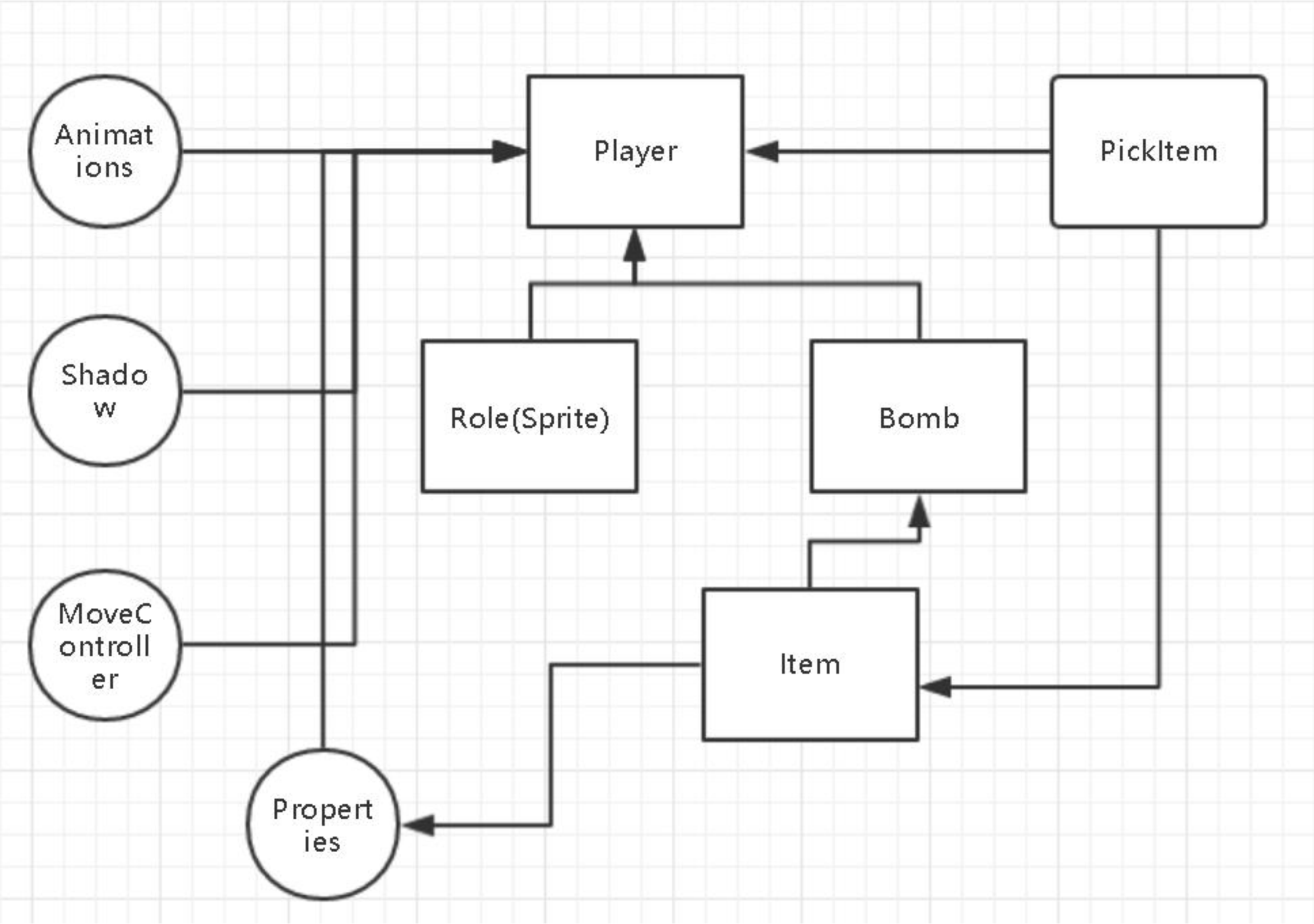
5.类关系图及描述

（1）场景类关系



所有的场景和弹窗都继自Cocos的Layer

（2）游戏中的各类关系



Player包含了一个role的Sprite，并通过建立一个shadow的Sprite来给role了addchild后实现阴影，与之类似给Bomb创建阴影，角色的动画也在创建角色通过Animations进行存储（包括静止、移动、骑宠、死亡），在后面需要使时可以直接调用，MoveController包含了对角色移动的控制函数（坐标的变换，动画的放映），而Properties相关内容用于控制角色的属性，初始化角色时根据选择角色的不同设置不同属性，拾取到道具过后也会相应发生变化，Bomb与Player建立has-a关系，捡到对应道具是也会修改到Bomb中的炸弹计数和威力范围，Bomb类也会将炸掉的块坐标传递给Item类。

6.C++知识点

贯彻且合理地使用了下列 3 条以上的 C++11 或更高的 C++ 新特性：

- 类与继承（✓）（上方代码已有示例）
- has-a关系运用（✓）（上方代码已有示例）
- 构造函数与析构函数（✓）
- new与delete（✓）
- 初始化列表（✓）

```
//Item.h
Item::Item(int &itemno, const cocos2d::CCPoint &TPos, cocos2d::CCTMXTiledMap* m_Map)
:tilePos(TPos), Map(m_Map),itemNo(itemno)
```

- 类型推断 (auto / decltype)（✓）(上方scene核心代码部分已有示例)
- 基于范围的 for 循环（✓）

```
//Player.cpp
void Player::dropBomb()
{
    if (!killedOrNot())
    {
        bool empty = true;
        auto roleTileCoord = tilecoordForPosition(role->getPosition());
        for (auto it : m_Bombs)
        {
            if (it->droppedOrNot())
                if (roleTileCoord == tilecoordForPosition(it->bombOpenglCoord()))
                {
                    empty = false;
                    break;
                }
        }
        if (empty) {
            for (auto it : m_Bombs)
            {
                if (!it->droppedOrNot())
                {
                    it->dropBomb();
                    myPlayerInformation.bomb_x = roleTileCoord.x;
                    myPlayerInformation.bomb_y = roleTileCoord.y;
                    SimpleAudioEngine::getInstance()-
>playEffect("MusicSource/appear.wav");
                    break;
                }
            }
        }
    }
}
```

- 智能指针
- Assert的使用（✓）

```
//PlayScene.cpp
    //get objects layer
    CCTMXObjectGroup *objects = gameMap->objectGroupNamed("objects");
    CCTMXLayer *architecture = gameMap->layerNamed("architecture-real");
    architecture->setZOrder(1);
    CCTMXLayer *floatlayer = gameMap->layerNamed("architecture-float");
    floatlayer->setZOrder(888);
    CCAssert(objects != NULL, "ObjectLayer not found");
```

- 常量表达式 (constexpr)
- Lambda 表达式
- 右值引用
- 字符串字面量（✓）

```
//Room.cpp
void Room::update(float dt) {
    if (test_model == false) {
        if (myMapSelect != 0 && myRoleSelect != 0) {
            choose();
        }
        else {
            text->setText("请选择地图和不同角色");
        }
        if (recv_[0] == '1') {
            notification = "The other choose map1";
        }
        else if (recv_[0] == '2') {
            notification = "The other choose map2";
        }
        else if (recv_[0] == '3') {
            notification = "The other choose map3";
        }
        else if (recv_[0] == 'o') {
            notification = "Game Start";
        }
        else if (recv_[0] == '\0')
            notification = "No people.";
        text->setText(notification);
    }
}
```

- static_cast的使用（✓）

```
//Item.cpp
cocos2d::CCPoint Item::getItemPosition()
{
    float x = static_cast<float>(tilePos.x) * Map->getTileSize().width;
    float y = (Map->getMapSize().height - 1 - static_cast<float>(tilePos.y)) * Map-
>getTileSize().height;
    return cocos2d::Vec2(x, y);
}
```

5.开发日志及项目进度

Commits on Jun 21, 2017

Function Update0.9.5(Repair the bug of server,add auto_close server f…….

Dinghow committed 21 hours ago

1b8480e

Function Update0.9.5(Repair the bug of server,add auto-close server f…….

Dinghow committed 21 hours ago

053e4a1

Commits on Jun 20, 2017

Function Update0.9.1(Repair the bug of server)

Dinghow committed a day ago

9ed9bc9

Function Update 0.9.0(Add network function,achieve online game,add mo… …

Dinghow committed a day ago

3af0c79

Commits on Jun 15, 2017

Function Update 0.8.4(Finish victory judge,add game countdown,repair ……

Dinghow committed 2 days ago

Commits on Jun 14, 2017

Function Update 0.8.0(Add offline model,repair animation bug,wrap the……

Dinghow committed 3 days ago

6c97dad

Merge remote-tracking branch 'origin/hpc' …

Dinghow committed 3 days ago

炸弹相互引爆，人物间相互杀伤

SherlockHpc committed 4 days ago

Commits on Jun 13, 2017

Function Update 0.7.4(Redefine class Role to Player,add struct to sav… …

Dinghow committed 4 days ago

Commits on Jun 12, 2017

File Update(Merge Zhc,Hpc and Ydh's latest edition)

Dinghow committed 5 days ago

Merge remote-tracking branch 'origin/hpc' …

Dinghow committed 5 days ago

Merge remote-tracking branch 'origin/zhc' …

Dinghow committed 5 days ago

Function Update 0.7.0(Add Hall scene,repair bug of Login scene's butt… …

Dinghow committed 5 days ago

炸弹间的相互引爆

SherlockHpc committed 5 days ago

randomNum function for dropping items optimized; …

Pomevak committed 6 days ago

Commits on Jun 11, 2017

Function Update(Merge Zhc's and Hpc's branches to master,add more eff……

Dinghow committed 7 days ago

a46f5f7

Merge remote-tracking branch 'origin/hpc' …

Dinghow committed 7 days ago

爆炸对人物杀伤的修正

SherlockHpc committed 7 days ago

Commits on Jun 10, 2017

Function Update0.6.2(Combine Hpc's branch to master)

SherlockHpccommitted an hour ago

Commits on Jun 7, 2017

Function Update 0.6.0(Add bomb dropping and explosion function)

SherlockHpc committed 3 days ago

Commits on Jun 6, 2017

Function Update 0.4.8(Add about us scene,and add volume controller,fi……

Dinghow committed 4 days ago

Function Update 0.4.5(Add a new map,add role select function,and add ……

Dinghow committed 4 days ago

Commits on Jun 5, 2017

Update README.md

Dinghow committed on **GitHub** 5 days ago

Update README

Dinghow committed 5 days ago

Function Update 0.4.0(Add map select function and related scene,repai……

Dinghow committed 6 days ago

Commits on Jun 4, 2017

File Update(Add new map,modifiy the map1 with adding three more spawn… …

Dinghow committed 6 days ago

Commits on Jun 3, 2017

Update README.md

Dinghow committed on **GitHub** 7 days ago

Commits on May 31, 2017

File Update(Add project's sln file of VS,add css file of CocosStudio)

Dinghow committed 10 days ago

Commits on May 30, 2017

Update README.md

Dinghow committed on **GitHub** 11 days ago

Function Update 0.3.5(Add background music play and transfer,optimize……

Dinghow committed 11 days ago

Function Update 0.3.1 & File Update(Divide PlayScene.cpp in terms of … …

Dinghow committed 11 days ago

Function Update 0.3.0(Finish collision check,repair scheduler bug,add… …

Dinghow committed 11 days ago

Commits on May 29, 2017

File Update(Add float layer to the map)

Dinghow committed 12 days ago

Function Update 0.2.0(Repair the animation bug happened when two keys……

Dinghow committed 12 days ago

File modification(add some annotation about role class)

Dinghow committed 12 days ago

Function Update 0.1.8(Finish role control,add role class,add role pro… …

Dinghow committed 12 days ago

Commits on May 28, 2017

Function Update 0.1.5(Optimize role control,add exit function)

Dinghow committed 13 days ago

Commits on May 26, 2017

Function Update 0.1.0(Add role control and animation,optimize the view)

Dinghow committed 16 days ago

Commits on May 21, 2017

Function update(add map layer and object , recreat the play scene)

Dinghow committed 21 days ago

Commits on May 18, 2017

Function update 0.0.1(Add map loading function)

Dinghow committed 24 days ago

Commits on May 10, 2017

Document update

Dinghow committed on 10 May

cocos project files

Dinghow committed on 10 May

Picture resource 1.0.0

Dinghow committed on 10 May

April to May

Learn cocos2d-x and related knowledge

April 8,2017

Submit the application of project