Cocos2dx第13周实验报告

1. 参考资料
2. <http://docs.cocos.com/cocos2d-x/manual/zh/other_node_types/tilemap.html> （瓦片地图）
3. <https://blog.csdn.net/firseve/article/details/50789526> （TiledMapEditor）
4. 课件ppt
5. 实验步骤

**·**需求及流程：

1. 随机产生怪物并且怪物会向角色靠近；

2. 怪物碰到角色后，角色掉血，角色血量为空则播放死亡动画并解除所有事件；

3. 角色可以攻击怪物；

4. 使用tilemap创建地图；

5. 加分项：使用本地数据存储，记录打到的怪物数量，同时在游戏中显示打倒数量；

**·**详细实现过程：

首先，产生怪物函数，在每次需要产生怪物时调用该函数即可：

1. **void** HelloWorld::addMonster() {
2. auto monster = Factory::getInstance()->createMonster();
3. **float** x = random(origin.x, visibleSize.width);
4. **float** y = random(origin.y, visibleSize.height);
5. monster->setPosition(x, y);
6. addChild(monster, 3);
7. }

怪物移动实现，利用定时器，在每个单位时间内，使得怪物角色激动：

1. **void** HelloWorld::updateCustomMonsterMove(**float** dt) {
2. Factory::getInstance()->moveMonster(player->getPosition(), 0.5);
3. }

实现怪物与角色碰撞检测：在该函数中调用Rect对象的方法containsPoint()判断是否存在角色所在空间内与怪物重合，即可实现角色与怪物碰撞检测。

1. Sprite\* Factory::collider(Rect rect) {
2. **for** (auto i : monster) {
3. **if** (rect.containsPoint(i->getPosition())) {
4. **return** i;
5. }
6. }
7. **return** NULL;

碰撞检测函数使用如下，在该函数中，通过碰撞检测的返回结果判断角色的状态，若二者碰撞，则怪物死亡，角色血条减少，当血条过少时，角色死亡，游戏结束：

1. **void** HelloWorld::hitByMonster(**float** dt) {
2. auto factory = Factory::getInstance();
3. Sprite\* collision = factory->collider(player->getBoundingBox());
5. **if** (collision != NULL) {
6. auto decrease = ProgressFromTo::create(0.4, pT->getPercentage(), pT->getPercentage() - 10);
7. pT->runAction(Sequence::create(decrease, CallFunc::create([&] {
8. **if** (pT->getPercentage() == 0 && flag) {
9. flag = 0;
10. auto deadAnimation = Animation::createWithSpriteFrames(dead);
11. deadAnimation->setDelayPerUnit(0.12f);
12. auto deadAction = Animate::create(deadAnimation);
13. player->runAction(deadAction);
14. menu->setEnabled(**false**);
15. unschedule(schedule\_selector(HelloWorld::updateCustomMonsterMove));
16. time->setString("Game over!");
17. unschedule(schedule\_selector(HelloWorld::updateCustom));
18. }
19. }), NULL));
20. factory->removeMonster(collision);
21. addMonster();
22. }
23. }

同时，增加一个定时器，时刻检测当前是否存在怪物与角色碰撞：

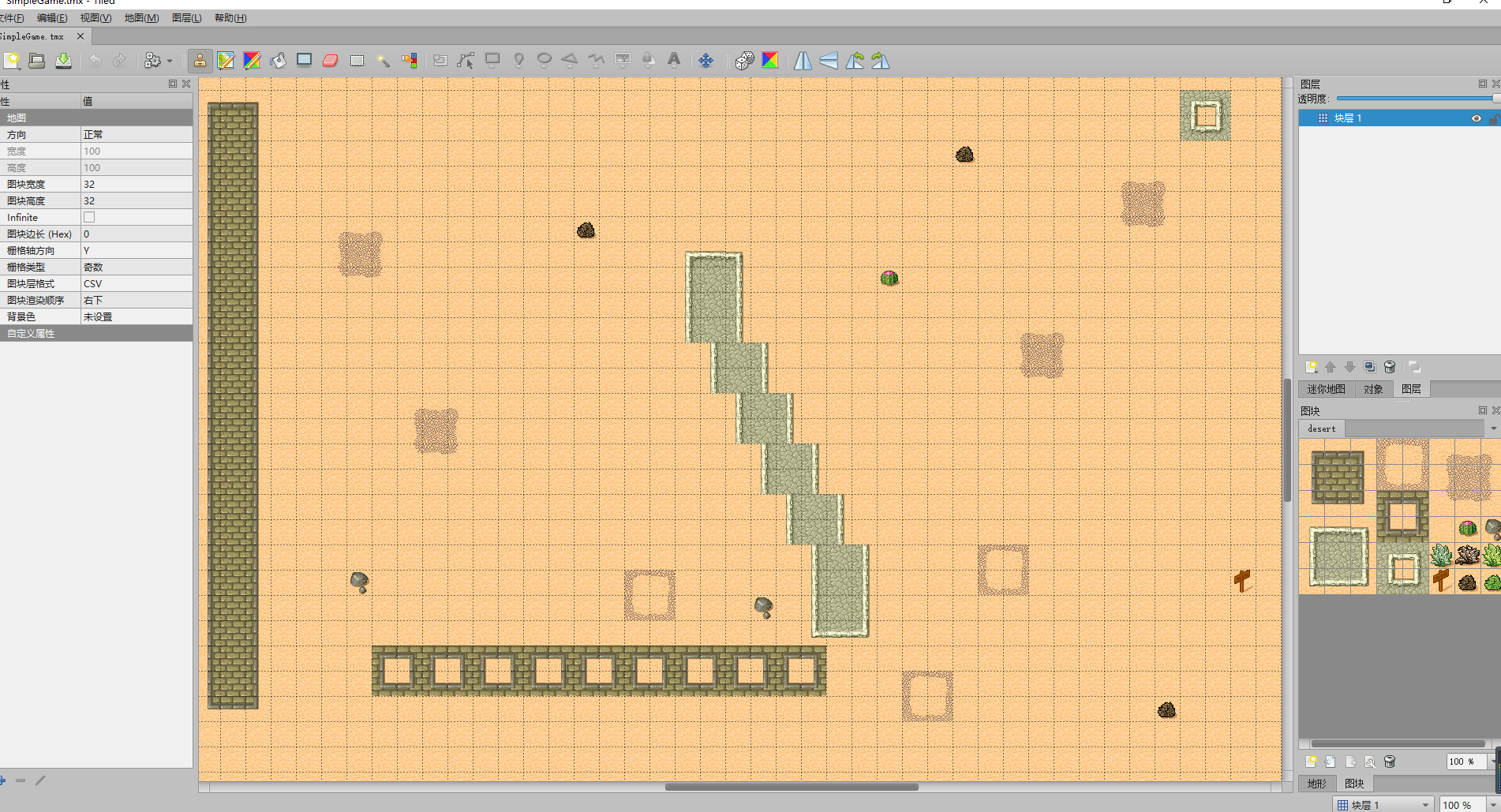
1. schedule(schedule\_selector(HelloWorld::hitByMonster),0.1f,kRepeatForever,0);

每0.1秒检测一次，可以确保在短时间若多个怪物与角色碰撞，则所有怪物都死亡。

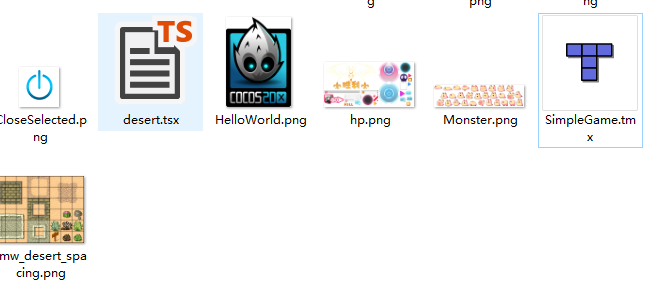
拓展YButton的功能，在攻击时，加上判断攻击范围内是否存在怪物，若存在怪物，则使其中一只怪物受到攻击而死亡；函数中通过设置攻击矩形区域，调用Factory的collider方法来实现：

1. auto YButton = MenuItemLabel::create(Y, [&](cocos2d::Ref\* sender) {
2. **if** (flag) {
3. flag = 0;
4. auto attackAnimation = Animation::createWithSpriteFrames(attack);
5. attackAnimation->setDelayPerUnit(0.12f);
6. attackAnimation->setRestoreOriginalFrame(**true**);
7. auto attackAction = Animate::create(attackAnimation);
8. player->runAction(Sequence::create(attackAction, CallFunc::create([&]{
9. flag = 1;
10. }), NULL));
12. Rect playerRect = player->getBoundingBox();
13. Rect attackRect = Rect(playerRect.getMinX() - 40, playerRect.getMinY(), playerRect.getMaxX() - playerRect.getMinX() + 80, playerRect.getMaxY() - playerRect.getMinY());
14. auto collision = Factory::getInstance()->collider(attackRect);
15. **if** (collision != NULL) {
16. auto increase = ProgressFromTo::create(0.4, pT->getPercentage(), pT->getPercentage() + 20);
17. pT->runAction(increase);
18. score++;
19. scoreLabel->setString(std::to\_string(score));
20. db->setIntegerForKey("score", score);
21. Factory::getInstance()->removeMonster(collision);
22. addMonster();
23. }
24. }
25. });

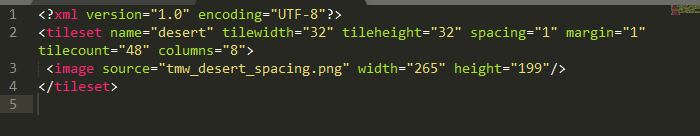
利用TiledMapEdito软件制作简易地图如下：



将地图所用元素源文件（png）和地图tsx文件模块以及tmx地图文件放置于Resources文件夹下：



在desert.tsx文件中制定了源文件的路径：



引用Tiledmap如下，与普通Node元素创建方法类似：

1. TMXTiledMap \*SimpleGameTMX = TMXTiledMap::create("SimpleGame.tmx");
2. SimpleGameTMX->setPosition(visibleSize.width / 2, visibleSize.height / 2);
3. SimpleGameTMX->setAnchorPoint(Vec2(0.5, 0.5));
4. SimpleGameTMX->setScale(Director::getInstance()->getContentScaleFactor());
5. addChild(SimpleGameTMX, 0);

本地数据存储创建如下，首先判断当前数据文件是否存在，如不存在则创建相应数据项，否则加载score到系统变量中：

1. db = UserDefault::getInstance();
2. **if** (!db->getBoolForKey("isExist")) {
3. db->setBoolForKey("isExist", **true**);
4. db->setIntegerForKey("score", score);
5. //db->setIntegerForKey("dtime", dtime);
6. }
7. **else** {
8. score = db->getIntegerForKey("score", score);
9. //dtime = db->getIntegerForKey("dtime", dtime);
10. }

在每次获得分数时，更新数据文件中：

1. scoreLabel->setString(std::to\_string(score));
2. 关键步骤截图

第一局获得17分：



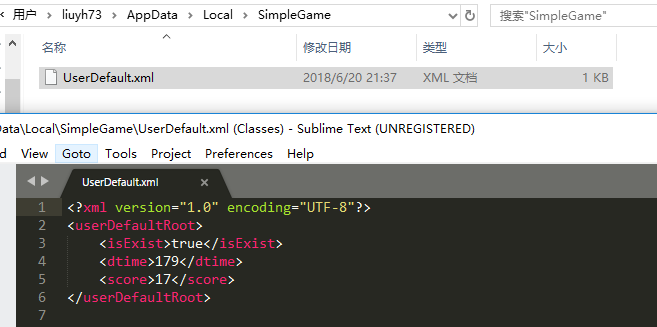


第二局打开时，分数初始值17分：



1. 亮点与改进（可选）

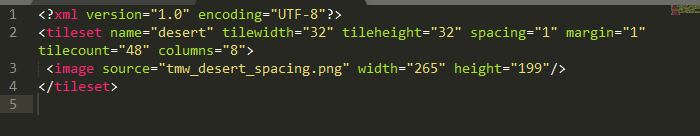
利用本地数据文件存储所得分数，并在屏幕上有所展示。



在系统文件夹C:\users\<username>\AppData\Local\SimpleGame\下有一个名为UserDefault.xml的文件，里面存储数据如上图（忽略dtime）。

1. 遇到的问题

问题一：

在创建瓦片地图时，若引用的原地图图片不再Resources文件夹下时，需要将原地图图片放置到Resources文件夹下，tsx和tmx也放置在该文件夹下，并且修改tsx文件，将

Image中source属性改为当前文件夹（Resources）中的图片即可。

问题二：

当角色血条少于0执行死亡动画时，若有其他monster碰撞角色，使得角色多次执行该动画，可以利用上节X和Y按钮的互斥来解决该问题：互斥变量。下属代码flag即为互斥变量。

