迷宫探险---研究与开发实践课程报告

程锦国 2019141460155

目录

[一、 游戏概述 1](#_Toc90898947)

[1.1 游戏整体模式 1](#_Toc90898948)

[1.2 游戏开发环境 2](#_Toc90898949)

[1.3 代码结构 2](#_Toc90898950)

[二、 游戏界面 2](#_Toc90898951)

[三、 游戏元素 3](#_Toc90898952)

[3.1 主人公 3](#_Toc90898953)

[3.2 地图 4](#_Toc90898954)

[3.3 忍者 4](#_Toc90898955)

[3.4 道具 5](#_Toc90898956)

[四、 游戏逻辑 6](#_Toc90898957)

[4.1 主人公与忍者的移动 6](#_Toc90898958)

[4.2 金币与药品刷新 7](#_Toc90898959)

[4.3 怪物模式主人公伤害逻辑 8](#_Toc90898960)

[五、 总结 8](#_Toc90898961)

# 游戏概述

## 游戏整体模式

游戏借鉴了泡泡堂等二维棋盘游戏以及迷宫游戏。程序利用深度优先算法，初始化生成一个迷宫，该迷宫只有唯一一条路通向终点，迷宫会随机生成数条分路来增加游戏难度。玩家只能看到迷宫的局部（如迷宫总规格未15\*15，玩家的视野只能探测到上下左右方向长度为4，当然这个长度可以依照实际情况做出调整），玩家需要用键盘（WASD）方向控制人物的运动，人物到达终点后游戏结束。

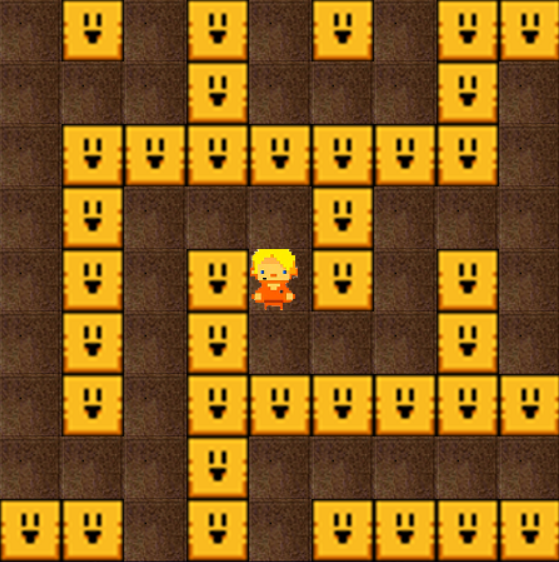


图1：普通模式操控界面

为了增加游戏难度，丰富游戏趣味性，我将设置四个不同的模式，不同模式之间的难度呈递增关系。

普通模式：在正计时情况下玩家走完迷宫成功到达终点

极限模式：玩家在倒计时结束前到达终点，迷宫中设置奖励道具可延续计时，考虑设置惩罚道具加快计时，玩家需在计时结束前到达终点。

金币模式：玩家须在倒计时结束前吃到足够多的金币并且到达终点，否则视为游戏失败。

恐怖模式：玩家须躲避迷宫中的怪物，在倒计时结束前到达终点。

不同模式搭配不同的背景音乐，给玩家带来更棒的游戏体验。

## 游戏开发环境

VS2013+easyx库

## 代码结构

c++做游戏的特点就是使用循环维持界面，但为了不让循环速度太快使界面崩溃，往往会在循环结束后加入睡眠语句，睡眠语句的长短由实际情况而定。这个游戏没有像跑酷游戏那么流畅，本游戏的主人公是一格一格走，对帧数要求比较低。所以选择每次休眠10毫秒。

最外层循环是主界面的模式选择。

接下来的循环是模式的外循环，负责初始化地图，时间，坐标，控制关卡，判断游戏的状态。

最内层的循环负责开始游戏后对地图，人物，地面，墙壁，怪物，道具的贴图。

# 游戏界面

开始界面采用贴图处理，并在上面添加了数段文本。以及四个按钮。



图2：游戏进入界面

四个按钮拥有焦点显示功能，即检测到鼠标移动到按钮上后，将按钮颜色由黑色修改为灰色，给用户及时反馈。

而玩家操控界面，玩家只允许拥有9\*9的视野。玩家需要在有限的视野下，纠正自己的路线，在有限的时间内，到达终点。在界面右侧有时间控制显示，以及一个按钮使玩家退出此模式到游戏主界面。



图3：游戏界面

# 游戏元素

## 主人公

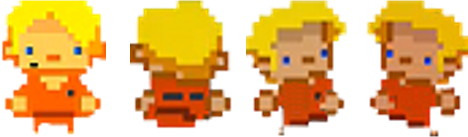


图3：主人公

主人公素材的选择来自研究与开发实践课中提供的素材，素材中提供了该人物移动方向的不同贴图。

主人公初始化于地图左上角，在二维数组中表示为（2，2）。人物的定位和移动由全局变量 x，y提供。并且在移动时读入方向信息，使屏幕拥有不同的贴图。

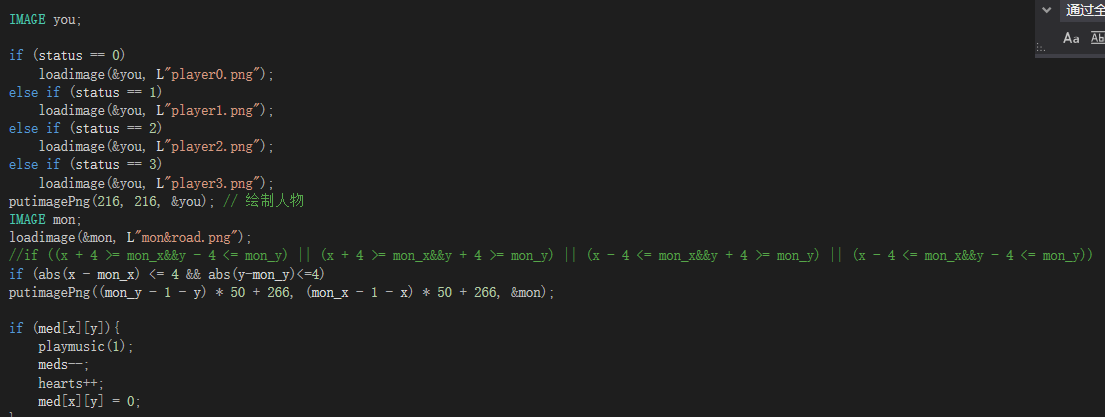


图4：人物移动贴图代码

## 地图

最简单的地图设计为15\*15，当然可以根据喜好任意扩充。值得一提的是，地图是使用深度优先搜索算法随机生成。

首先，全地图初始化为墙壁，规定地图最外层一圈为固定墙壁不可修改，然后程序从屏幕左上角出发，也就是主角的复活点，开始递归前进，前进方向随机决定，当碰到墙壁后返回，当相邻的点周围的所有点都为被标记过的，或与被标记的点相邻，则返回。直到能继续走为止。直到最后地图中没有符合条件的被标记的点。最后将地图右下角标记为迷宫出口。

由上可知，整个地图有且仅有一条道路通向出口。

## 忍者

怪物模式中的忍者，随机降临于某一个地点，阻止主人公走出迷宫。忍者的移动模式与主人公不同，忍者的贴图只有一种，因为没有找到合适的素材，但实际应用后发现单一贴图更符合忍者冷酷的外表。由于单线程，所以忍者只能与玩家同时移动，但这并没有影响游戏的可玩性。



图5:忍者贴图

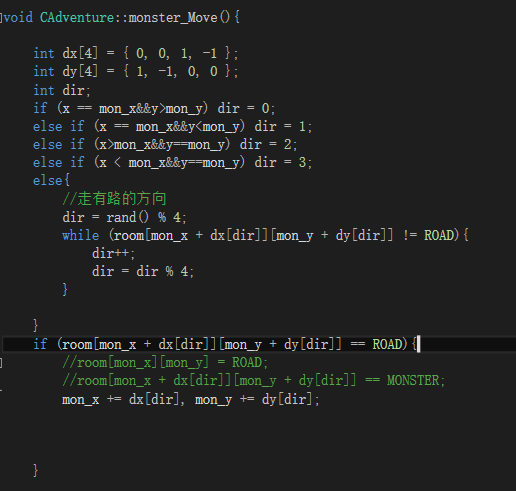


图6：忍者移动代码

忍者的属性是神出鬼没，移动速度极快，游戏中设置了忍者每次最多可移动四个单位距离。并且，尽管随机生成在地图的某一点，忍者拥有感知能力，可以刚知道如果主人公与它处于一条直线上时，便朝着这个方向追杀主人公，但不在一条直线时，随机选择有路的方向前进。并且不需要完全接近主人公，在距离2以内即可对主人公造成伤害。

## 道具



图7：金币与药品道具

金币模式中，主人公需要收集足够多的金币，才能成功通关，否则视为游戏失败。在怪物模式中，主人公可以通过吃散落在地图中的药品来给自己回血，每一个药品回一血，最多5血。

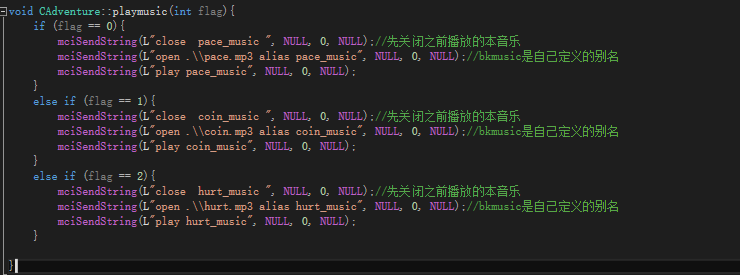
## 音乐

游戏音乐分为背景音乐与音效。

对于背景音乐，不同的模式根据其氛围给予不同音乐。对于普通模式，急速模式，金币模式，因其模式内容比较轻松，所以设定一个比较轻松的背景音乐。对于怪物模式，给了一个比较紧张的背景音乐。

图8：模式背景音乐

对于音效，总共有三种，人物脚步，吃金币和药品，被忍者追杀受伤害。

****

# 游戏逻辑

## 主人公与忍者的移动

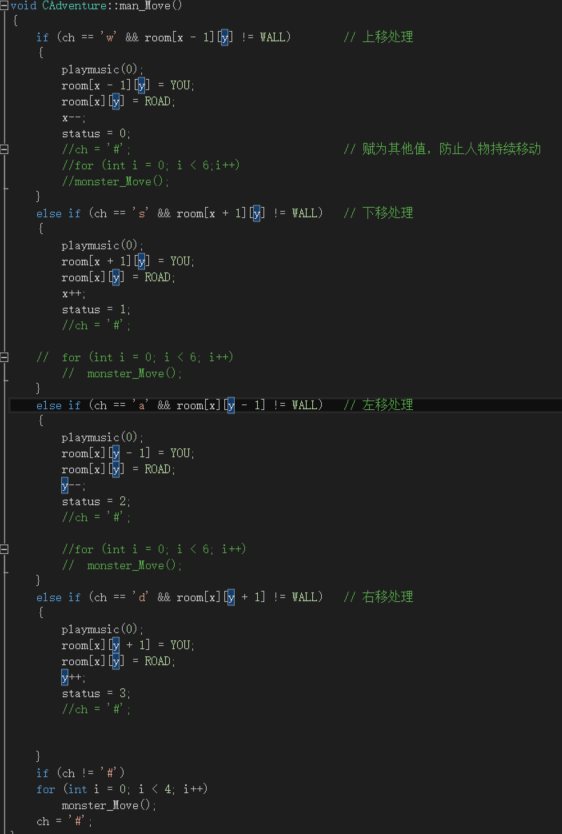


图8：人物移动逻辑

人物通过变量ch获取键盘信息，并且为了防止持续移动，每次移动后将ch置为‘#’，并且在移动前判断前方是否为路，否则不予移动。

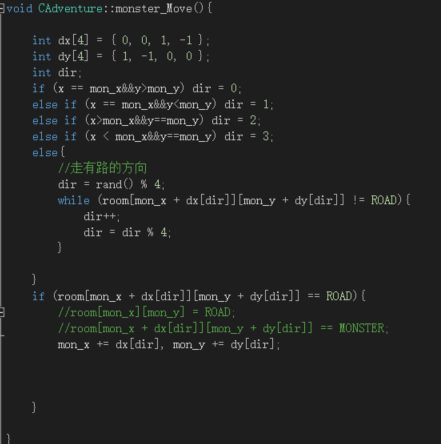


图9：忍者移动逻辑

忍者的移动的执行随主人公的移动而执行，并且当它与主人公处于同一直线，向着忍者方向移动否则随机选择可以移动的方向移动。随着主人公移动一次，它移动四次，这样做的目的是，提高忍者的跟踪及猎杀效率，增加游戏的难度。

## 金币与药品刷新

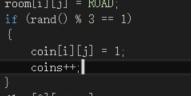


图10：金币刷新逻辑

金币与药品的刷新逻辑为在地图生成过程中，每当一个点为道路时，金币三分之一概率刷新，药品十分之一概率刷新。

## 怪物模式主人公伤害逻辑

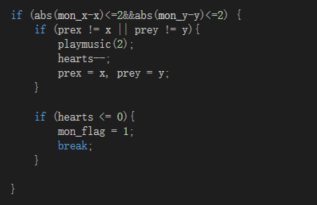


图11：主人公受伤逻辑

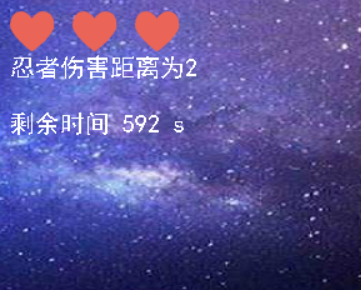
主人公在忍者距离为2的范围内，收到一点伤害，hearts减一。其中hearts表示生命数量（血量）显示在界面右上角。

图12：血量显示

# 总结

总的来说，本游戏的逻辑设计和界面设计相当粗糙。本游戏因为是二维迷宫游戏，所以游戏有意思的玩法比较少，而且新增道具比较困难，因为该游戏的逻辑较为简单，即使主人公到达终点，在这种情况下，阻止玩家顺利通过终点的游戏玩法比较有限。

游戏的迷宫自动生成不同于其他游戏需要作者手动设置就可以自动生成，而且可以按照自己喜好扩大地图，增加游戏难度，可以算作本游戏的创新点所在。

我在此次研究开发实践课中，感受颇深，无论是游戏还是软件，首先做的是找一个好点子，并证明其可行性，其次是选择一个合适的框架，并掌握它，所幸本节课使用的easyx框架提供的接口函数都比较简单，对新手特别友好。掌握框架函数是开发的一个最重要的事情，因为如果直接上手开发，往往不知道如何使用某个接口函数，甚至不知道解决一个问题需要用到哪个函数。明白这个道理会对我日后的软件开发提供巨大的帮助。