32 组软件工程大作业设计总报告

组员：惠炳谕 黄河源 刘杨 张添龙 李珺
分工：素材搜集：黄河源 刘杨
代码实现：惠炳谕
软件测试：黄河源
报告撰写：李珺
剧本编辑：惠炳谕 黄河源 李珺

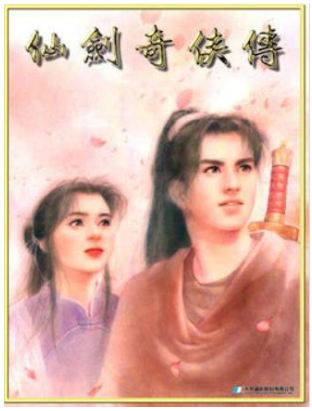
一、需求分析

《西中战记》
需求规格说明书

作者：惠炳谕 黄河源 刘杨 张添龙 李珺
组别：32 组

1、需求分析

说起早期，中国也有辉煌的游戏历史



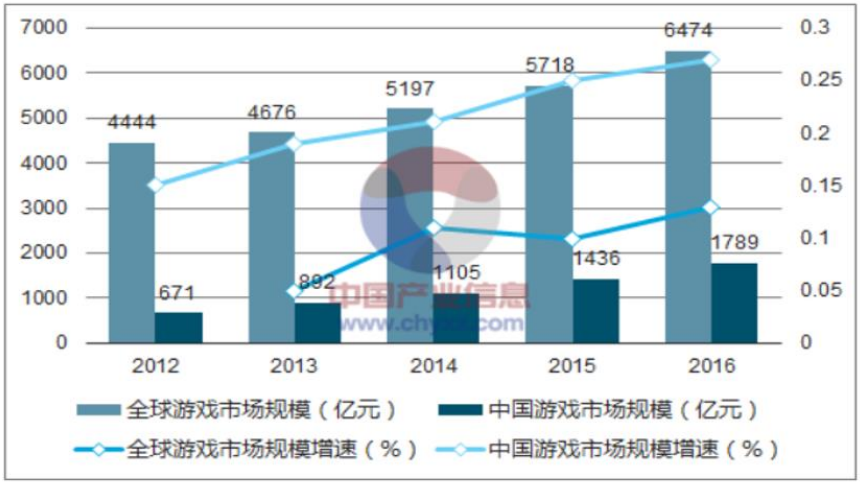
也有我们小时候的回忆

现在游戏行业也在蓬勃发展




近几年来中国以及世界的游戏市场规模越来越庞大，带动了巨大的资金流动

2012-2016 年中国和全球游戏市场规模



根据腾讯 2016 年财报显示，《王者荣耀》日活跃用户超过 5000 万，包括王者荣耀在内的移动游戏总收入达到 384 亿元。

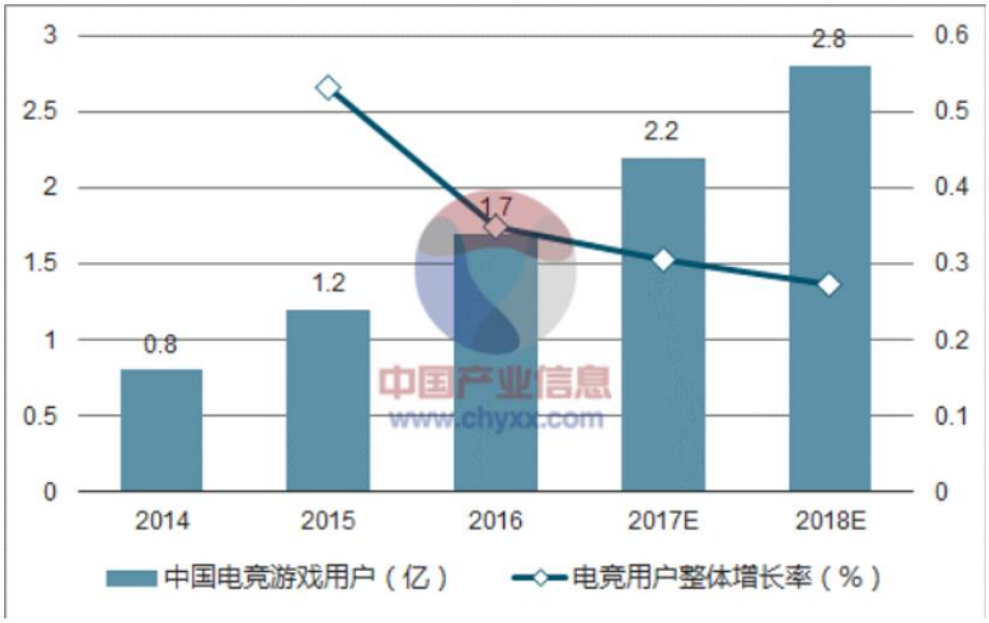
今年一季度，腾讯控股总收入约 496 亿元，同比增长 55%。其中，移动手游营收达到 129 亿元，同比增长 57%。

王者荣耀已然成为了一款全民游戏，根据极光大数据显示，2017 年 5 月《王者荣耀》的用户规模超过 2 亿，日活跃用户达 5412.8 万人，月活跃用户达 1.63 亿，而这个数字较去年 12 月增长了一倍。

王者荣耀月收入 30 亿，登顶全球第一

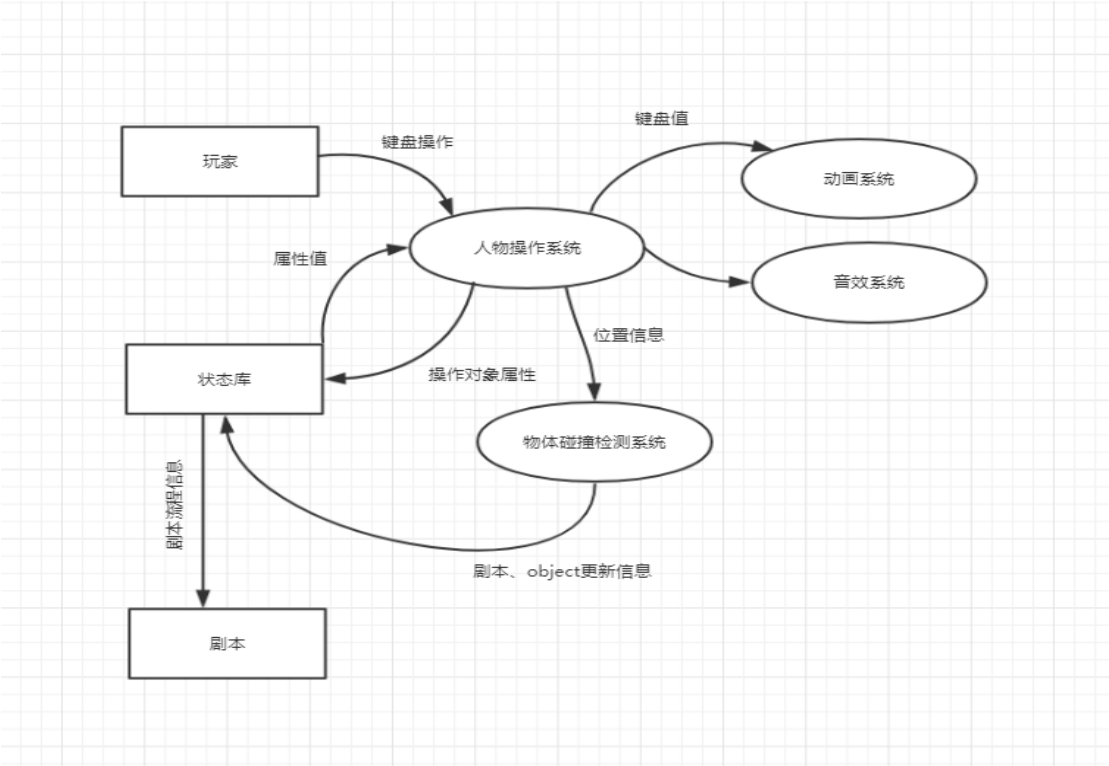
近几年的游戏也正在往职业化发展：

游戏似乎越来越向着一个行业，一个职业化方向发展

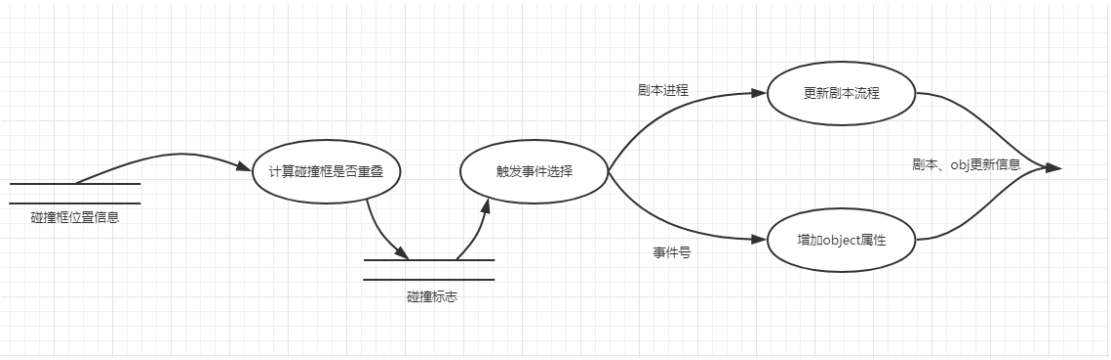


借用围城的一句话来说：
围城里的人并不想出去，围城外的人还在不断的进来
在压力巨大的社会以及越来越向安逸型社会发展的趋势下，人们日常的娱乐需求越来越得不到满足，可以说，哪里有游戏，哪里就有商机！人们的需求甚至在向着一个越来越难以满足的方向发展。

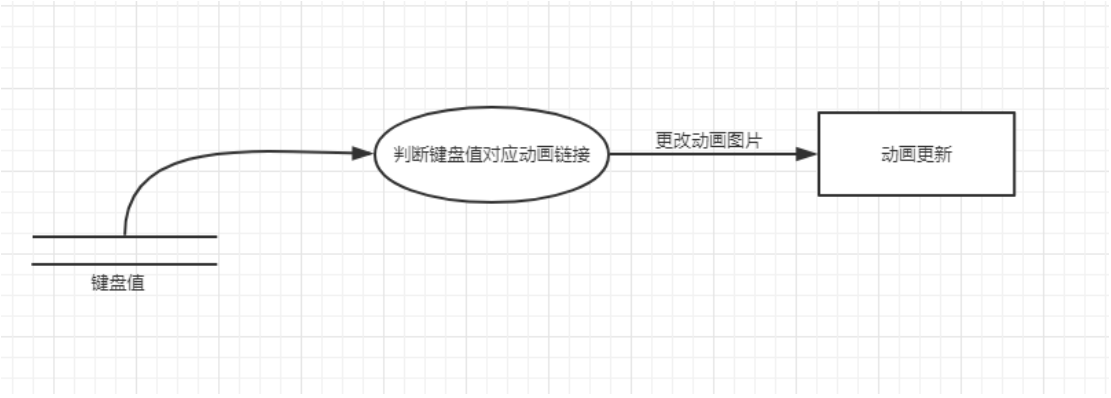
2、数据流图



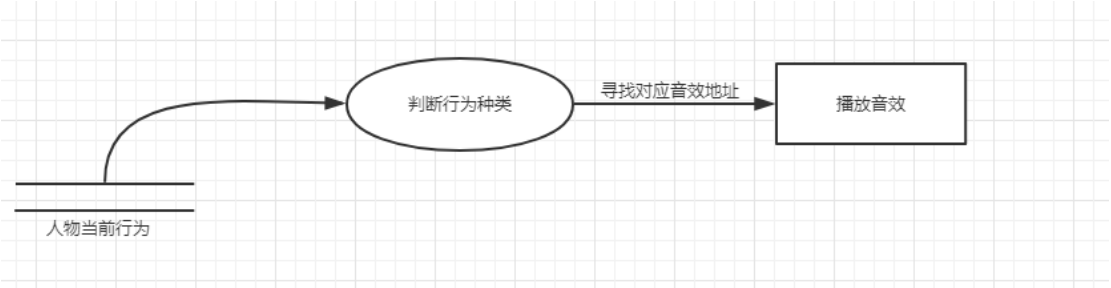
顶层图



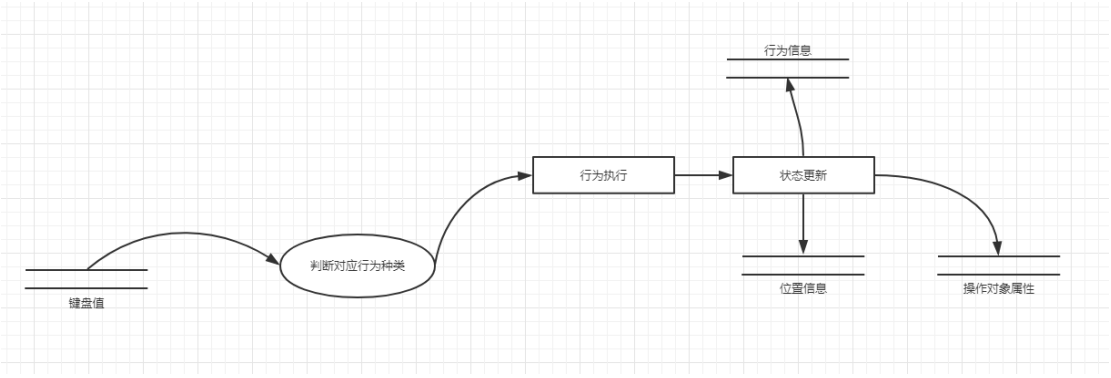
碰撞检测系统分层数据流图



动画系统分层数据流图



音效系统分层数据流图



人物操作系统分层数据流图

数据字典

数据文件：状态库

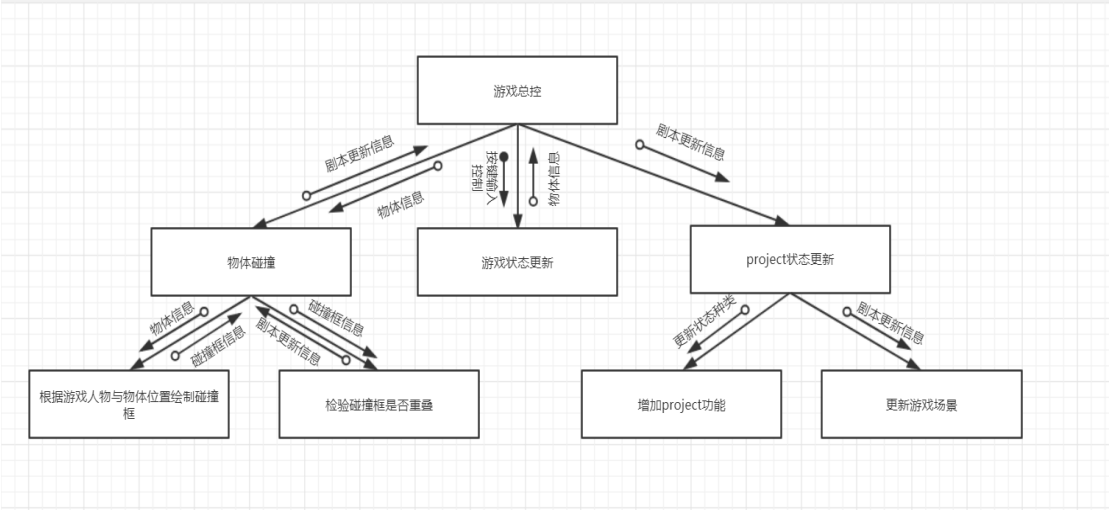
文件组成：位置信息，操作对象属性、键盘值、碰撞标志、事件号、剧本进程信息

数据项：位置信息

数据类型：浮点型
数据项：位置信息
数据类型：字符型
数据项：键盘值
数据类型：字符型
数据项：碰撞标志
数据类型：布尔型
数据项：事件号
数据类型：整型
数据项：剧本进程信息
数据类型：整型
数据流定义
键盘值
名称：键盘值
简述：读取从键盘中键入的键对应的数值
数据来源：外部设备
数据去向：人物操作系统判断模块
数据组成：键盘按键对应字符值
位置信息
名称：位置信息
简述：反馈游戏内各个 object 的位置信息
数据来源：系统界面、按键值计算结果
数据去向：碰撞检测系统
数据组成：玩家位置
玩家碰撞框左上角坐标与右下角坐标
物体位置
物体碰撞框左上角坐标与右下角坐标
操作对象属性
名称：操作对象属性
简述：返回玩家操作的对象目前的等级、功能项
数据来源：人物操作系统
数据去向：状态库
数据组成：与 npc 对话状态
人物技能
人物目前形态
人物当前拥有物品

二、概要设计

软件结构图：



概要设计说明书

以我设计的游戏为例，我本次设计的游戏是一个解密风格类游戏，最主要的是整个游戏随着人物 object 与重要事件碰撞而触发游戏事件，更新游戏剧本，进而随着剧本线性的推进到达游戏结局，其中可将整个系统分为碰撞模块，游戏流程状态即进度模块以及游戏 object 模块，这里做解释，object 即为一个游戏包括操作人物在内一切可产生交互的物体。
其中物体碰撞模块为只要检测到符合条件的两个 object 产生了碰撞事件，那么我们就会更新剧本流程，推进任务主线。而推进任务主线代表着我们需要更新剧本信息，更新 object 状态属性，如游戏人物在奇遇后学会特殊技能、接收信息提示，人物是否满足学习的等级等等。
图中我们可以看到在检测物体产生碰撞事件后更新剧本、更新 object 状态等等的顺序。

软件详细设计说明书

本次软件设计为一个解密类 exe 格式的电脑游戏，玩家可通过 W、A、S、D 操纵人物移动，并在游戏中探索，最终达成最终目标，下面对整个游戏的流程进行说明
1、初遇
游戏玩家初入游戏，系统开始播放背景音乐，并弹出介绍框为玩家介绍游戏


迷之声音：
WASD 键分别控制左右移动。嗯，就这样。
啥？我骗你感情？不不不，那只是你内心
最真实的想法而已
请按空格键继续


迷之声音：
走到妹子旁边按 G 键即可搭讪（手动滑稽）
请按空格键继续

介绍完成后我们玩家开始操纵人物根据提示开始解密，首先我们走到蓝色 NPC 位置与其对话获得信息，对话到一定程度我们知道了玩家的功能按键

许多看似无用的信息在解密游戏中起到至关重要的作用

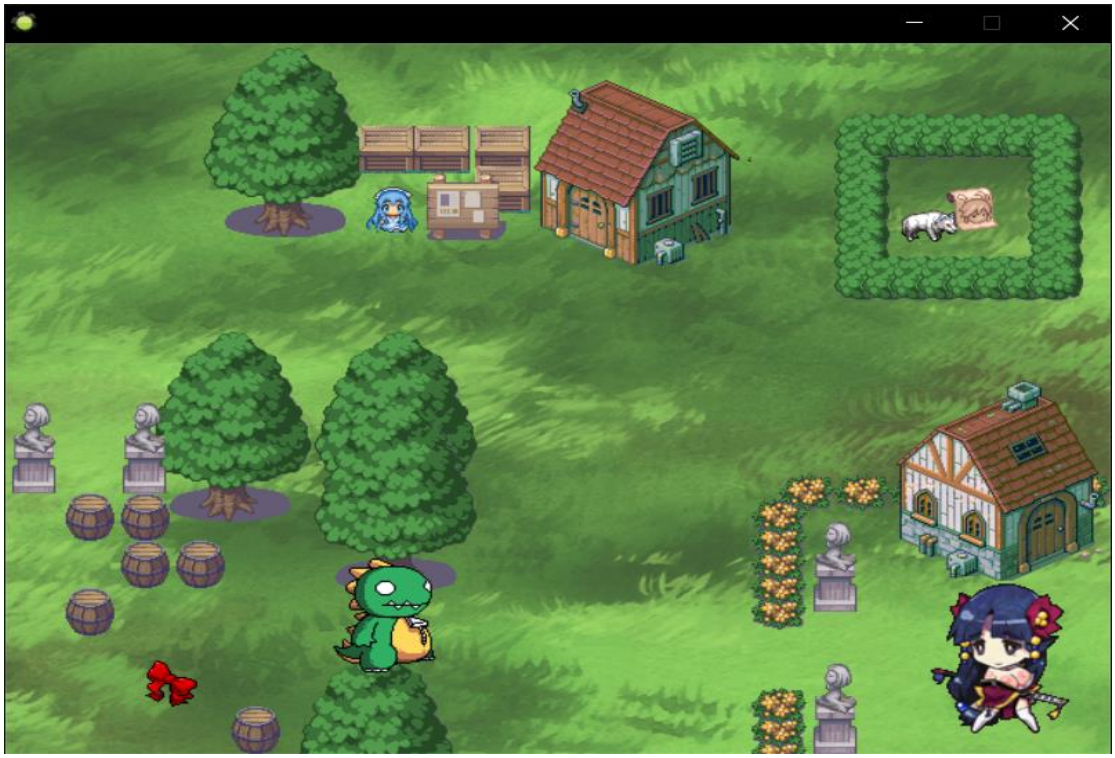
获得任务目标

当然也会有任务提示

解密游戏中收集场景元素是必不可少的一环，我们尝试去拿地图左上角的卷轴会发现无法穿

过草丛，但我们可以从旁边的提示牌得到提示

按 K 键变成狗后可以获得卷轴



我们尝试用新学的技能攻击守关怪兽，瞄准怪兽按 I 键发动攻击

但获得提示提醒技能不够强力，说明我们需要去别的地方寻找信息，此时地图上唯一没有被我们探索过的地方就是右下角的人物，我们去尝试触发事件

这时候我们可以回想到蓝发 NPC 说过的看似无用的话，实则暗藏答案

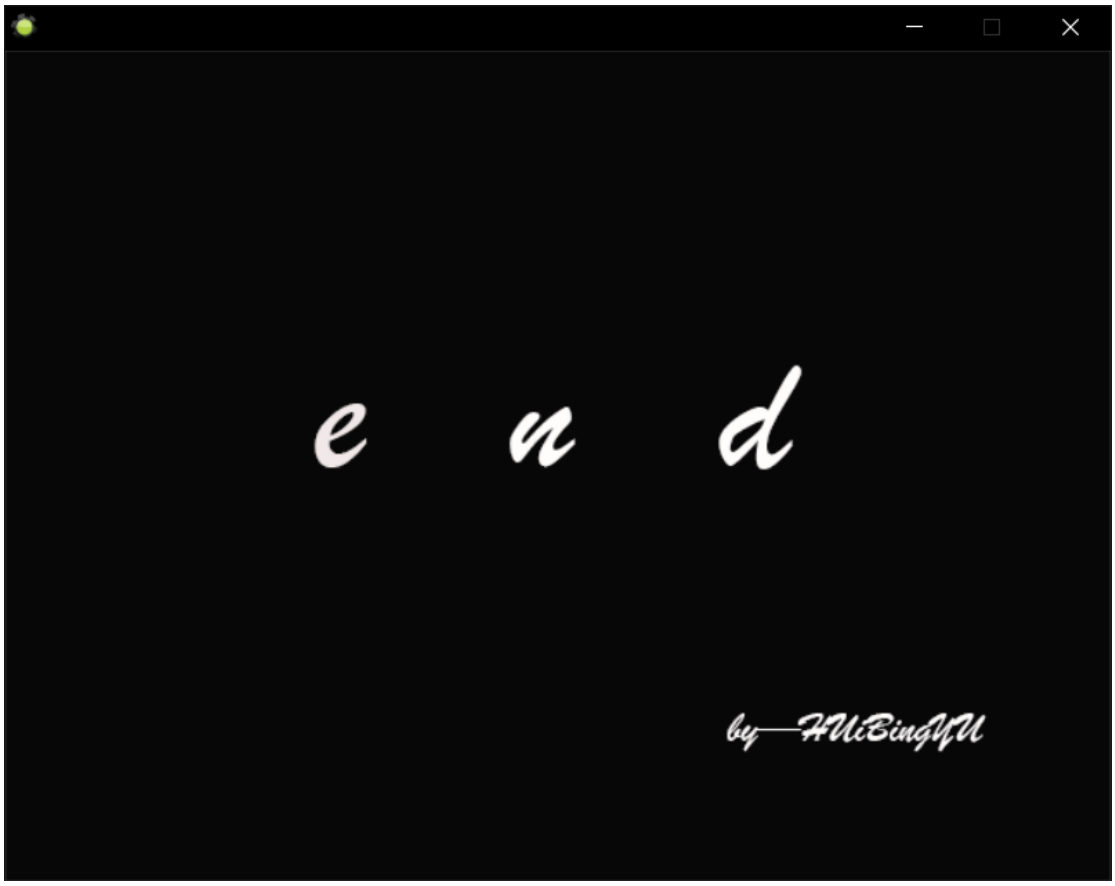
我们利用狗状态与人物对话

此时我们再次尝试用新技能攻击怪物

怪物被打倒，被阻挡的路畅通了

我们拿到钥匙

回去与 NPC 对话，游戏终。



整个软件其实并未涉及到攻击系统。玩家通过发射子弹产生物体并与场景物体 obj 碰撞出发事件，是一个伪攻击系统，整个游戏通过 object 与 object 的碰撞触发事件来推进剧本流程，最终到达结局。所以碰撞系统是重中之重。

软件测试

1：白盒测试（逻辑测试）

与 NPC 未碰撞是否可以触发对话	否	测试通过
人物是否可以在未变身的情况下进入草丛	否	测试通过
人物未获得卷轴前是否可以使用技能	否	测试通过
人物是否可以绕过怪兽进入森林	否	测试通过
人物在人类形态下是否可以获得神秘人物的奖赏	否	测试通过
人物在未获得钥匙的情况下是否可以触发结局	否	测试通过
人物是否会发生按键无法控制或控制混乱	否	测试通过
与 NPC 碰撞是否可以触发对话	是	测试通过
人物是否可以在变身的情况下进入草丛	是	测试通过
人物获得卷轴后是否可以使用技能	是	测试通过
怪物是否会被等级一的技能打败	否	测试通过
人物在动物形态下是否可以获得神秘人物的奖赏	是	测试通过
人物在获得钥匙的情况下是否可以触发结局	是	测试通过
获得神秘人物的奖赏后是否可以使用终极技能	是	测试通过
怪物是否会被终极技能打败	是	测试通过

2：黑盒测试（功能测试）

音乐是否正常播放	是	测试通过

人物移动是否正常	是	测试通过
是否无法穿过障碍物	是	测试通过
是否可以捡起钥匙	是	测试通过
是否可以穿过怪物	否	测试通过
碰撞是否会触发事件	是	测试通过
空格键是否会结束或推进对话	是	测试通过

测试说明：

最终对游戏的进程逻辑和基本功能进行了测试，期间出现了无法结局，无法对话，音乐只播放一次的 BUG 最终均得到修复。最终测试全部通过。由于游戏软件比较特殊，实在不知道该如何用书上的方法来测试，于是商讨后从黑、白盒测试的定义出发进行了测试，最终测试完成，游戏完美运行，由于整个游戏每一个物体都有对应的代码，综合下来篇幅较长，在此附上主角的控制代码，其余代码可通过 gamemaker 打开游戏文件（.gmk 后缀）看到。

创建代码

```
state = 0;//0 正常；1 死亡;2 武器
face = 0//3 向下；1 向上；2 向左；0 向右
timer=0;
image_speed = 0;
shoot_timer = 0;
```

步代码（人类形态）

```
var dx,dy,psp,tmp;
dx = 0;
dy = 0;
psp = 3;
//***按键检测
if(keyboard_check(ord('W')))
{
	face = 1;
	dy = -1;
}else
if(keyboard_check(ord('S')))
{
	face = 3;
	dy = 1;
}else
if(keyboard_check(ord('A')))
{
	face = 2;
	dx = -1;
}else
if(keyboard_check(ord('D')))
{
	face = 0;
```

```
        dx = 1;
}
//***发射
if(state==2){
      if(keyboard_check(ord('I'))&& shoot_timer==0&&instance_number(obj_bullet)<1)
      {
            shoot_timer = 10;
            tmp = instance_create(x,y,obj_bullet);
            tmp.direction = face*90;
            tmp.speed = 6;
            sound_play(sound3);
      }
      }
//***射击计时器自减
if(shoot_timer>0)shoot_timer -=1;

//撞墙检测
if(collision_rectangle(x-4+dx*psp,y-4+dy*psp,x+4+dx*psp,y+4+dy*psp,obj_wall_father,0,0))
{
      dx=0;
      dy=0;
}
if(collision_rectangle(x-4+dx*psp,y-4+dy*psp,x+4+dx*psp,y+4+dy*psp,obj_cao2,0,0))
{
      dx=0;
      dy=0;
}
//坐标限制
if(x-16<0) x= 16;
if(x+16>room_width) x=room_width-16;
if(y-16<0) y= 16;
if(y+16>room_height) y=room_height-16;



x += dx*psp;
y += dy*psp;

//***动画控制
if(dx!=0||dy!=0)
    image_index += 0.5;

//外观改变
switch(face)
```

```
{
case 3:sprite_index = spr_player_down;break;
case 1:sprite_index = spr_player_up;break;
case 2:sprite_index = spr_player_left;break;
case 0:sprite_index = spr_player_right;break;
}

//变身系统
if(state==0){
if(keyboard_check(ord('K')))
{
        instance_create(x,y,obj_bianshen);
        instance_destroy();
}
}
if(state==2){
if(keyboard_check(ord('K')))
{
        instance_create(x,y,obj_bianshen2);
        instance_destroy();
}
}
```

步代码（狗形态）

```
var dx,dy,psp,tmp;
dx = 0;
dy = 0;
psp = 2;
//***按键检测
if(keyboard_check(ord('W')))
{
    face = 1;
    dy = -1;
}else
if(keyboard_check(ord('S')))
{
    face = 0;
    dy = 1;
}else
if(keyboard_check(ord('A')))
{
    face = 2;
    dx = -1;
}else
```

```
if(keyboard_check(ord('D')))
{
        face = 3;
        dx = 1;
}
//撞墙检测
if(collision_rectangle(x-4+dx*psp,y-4+dy*psp,x+4+dx*psp,y+4+dy*psp,obj_wall_father,0,0))
{
      dx=0;
      dy=0;
}

x += dx*psp;
y += dy*psp;

//***动画控制
if(dx!=0||dy!=0)
    image_index += 0.5;

//***变身成人
if(state==0){
if(keyboard_check(ord('J')))
{
            instance_create(x,y,obj_player2);
            instance_destroy();
            obj_player2.state=2;
}
}
if(state==0)
{
      switch(face)
{
case 0:sprite_index = spr_player_dog_down;break;
case 1:sprite_index = spr_player_dog_up;break;
case 2:sprite_index = spr_player_dog_left;break;
case 3:sprite_index = spr_player_dog_right;break;
}
}
```