# 4计划

## 4.1总体设计

软件测试遵循十原则：1.测试应基于用户需求2.做好软件测试计划是做好软件测试工作的关键3.应尽早的开始软件测试并不断的进行软件测试4.测试前必须明确定义好产品的质量标准5.避免测试自己的软件6.应充分注意测试中的集群现象7.必须检查每个实际输出结果8.穷举测试是不可能的9.测试设计决定了测试的有效性和效率10.注意保留测试设计和说明文档，并注意测试设计的可重用性

测试主要是功能性测试，采用多运行顺序执行法。

### 4.1.1测试级

系统测试（System Test）：这是行业做的最多的一种测试，由测试人员充当用户，对软件的功能主体进行测试 。

### 4.1.2测试类别

可用性测试。

### 4.1.3一般测试条件

调低探照灯速度与伤害，使得过关难度降低的基础上，进行我们的测试工作。关于具体的数据设置等到测试完功能后进行实际上手的体验做出倾向于游戏性上的微调。

### 4.1.4测试过程

第一次测试，故意暴露在灯塔的探照灯下，看角色是否会阵亡，并重生在起点。

第二次测试，在慢速灯光下先后规避两个灯塔的探照灯，到达检查点，然后顺利通过第一部分的测试。

第三次测试，在慢速灯光下先后规避两个灯塔的探照灯，到达检查点后回到第二个灯塔下故意触碰灯光，看是否能够重生在新的检查点处。

第四次测试，通过跳上路灯走隐藏道路，查看是否可以不规避探照灯便成功通过该部分。

### 4.1.5数据记录、归约和分析

经测试发现如果不对角色范围加以限制，角色会掉出地图外且不阵亡重生，从而陷入不断下落的错误状况，经过调整，将角色可以跳跃出的地方封死。

关于探照灯的移动速度问题，加快探照的移动速度可以增加难度，而难度的增加在一定程度上能够增加游戏性，增长游戏游玩时间。

## 4.2计划执行的测试

我所负责的是第一关第一部分各功能能否正常运行的测试，主要分为以下几条：

1. 主人公能否正常被键盘控制
2. 主人公被灯塔灯光照射后是否判定死亡
3. 主人公能够正常通过灯塔
4. 检查点是否正常工作

### 4.2.1第一关第一部分

#### 4.2.1.1人物操纵与第一部分

a.测试对象；游戏原型

b.测试级；系统测试，由开发人员充当用户

c.测试类型或类别；可用性测试

d.需求规格说明中所规定的合格性方法；

e.本测试涉及的CSCI需求和软件系统需求的标识符；test\_001

f.特殊需求；人物操作需要键盘实时键入

g.测试方法，依照经验进行的可用性测试

h.要记录的数据的类型；卡关的对应操作，与探照灯移动速度数值。

i.要采用的数据记录/归约/分析的类型；无

j.假设与约束，如由于系统或测试条件即时间、接口、设备、人员、数据库等的原因而对测试产生的预期限制；要求可以在个人电脑上顺利且流畅进行

k.与测试有关的安全性、保密性与私密性要求：保证游戏原型的安全性，不会引起系统崩溃。

## 4.3测试用例

a.测试用例的名称和标识；test\_case\_001

b.简要说明本测试用例涉及的测试项和特性；测试项主要是人物操纵与关卡设计。特性是测试结果一目了然，但是却不能量化地进行测试。

c.输入说明，a人物左移，d人物右移，空格跳跃。

d.输出说明，根据键盘输入的不同，人物做出不同的动作。可以组合，例如移动中跳跃。

e.环境要求，个人电脑，unity引擎。

-------------------------------------------------------------------------------------------------------------------------------

# 7评价

## 7.1评价准则

关于第一部分关卡的测试工作，我们主要采用了可用性测试的方法，这是出于项目本身测试起来实时性很强，且直接体现在游戏运行的结果上。虽然这种测试效果比较直观，但是也具有不小的缺点，它的输入输出并不复杂，并且不能量化地进行测试，不能大规模进行测试。

## 7.2数据处理

本部分的测试设计到的数据处理，主要是操纵人物时的键盘输入与探照灯移动速度数值上的调整。其中操纵人物时的键盘输入主要考虑不同输入的连续、同时输入是否会造成不良的后果与错误，探照灯移动速度的调整则主要考虑关卡难易度与游戏性上的问题。

## 7.3结论

本次的测试总的来说比较简陋，且由于可用性测试的特性，在测试过程中实时凭经验进行整个测试过程，相较而言，整个测试的量级也比较的小，并且不能使用机器量化地分析代码进行测试，而是要耗费人力。

但好在由于本项目团队五人，工程量并不大，在测试方面使用机器测试反而显得冗余累赘，本部分的测试便显得更为轻便。

-------------------------------------------------------------------------------------------------------------------------------

# 3测试结果概述

## 3.1对被测试软件的总体评估

通过本次的测试，我们发现人物操作上没有明显的问题，但是在关卡设计上有一定的问题，比如玩家可以向地图的左边走，从而跳出地图外不停下坠，并卡在此状态无法结束，解决方法是添加了一些砖块形成一个高台，使得玩家无法跳跃到高台上，从而无法到达地图的另一边。此外，游戏关卡部分的探照灯速度较快，使得玩家需要反复尝试才能通过，为了体验着想我们降低了其速度数值。另外，关于在路灯间跳跃以规避探照灯的设计，由于路灯间间隔较短，在跳跃时还需要控制不让角色跳过头从而掉下路灯，与本身设计这个容易通过的隐藏通道的初衷不一致，故增大间距，使得可以直接跳到下一个路灯上。以上便是测试的大致过程。

## 3.2测试环境的影响

在不同个人电脑上的同一个版本号的unity打开同一个unity项目时，部分电脑会报错，无法打开，后经研究发现是使用GitHub同步时使用gitignore文件忽略本地打开项目产生的临时文件造成的影响，故不能打开者删除library文件夹后git pull即可打开项目。

## 3.3改进建议

添加了一些砖块形成一个高台，使得玩家无法跳跃到高台上，从而无法到达地图的另一边。降低了探照灯移动速度数值。增大隐藏通道中路灯的间距。

# 4详细的测试结果

4.1 test\_001

4.1.1测试结果小结

本次测试遇到了一些问题，并进行了及时的修正。

4.1.2遇到了问题

4.1.2.1 test\_case\_001

a.所遇到问题的简述；角色可以从左边走出地图外

b.所遇到问题的测试过程步骤的标识；进入游戏后按a键向左移动。

c.对相关问题/变更报告和备份数据的引用；在原有项目2d-game上进行小幅度修改

d.试图改正这些问题所重复的过程或步骤次数，以及每次得到的结果；在左边新增一些地图块阻碍角色跳下地图块。

e.重测试时，是从哪些回退点或测试步骤恢复测试的。由于是第一部分测试，从起点开始。

4.1.3与测试用例/过程的偏差

4.1.3.1 test\_case\_002、test\_case\_003

a.偏差的说明:两个测试实例中探照灯的移动速度不同。

b.偏差的理由；通过不同移速的测试案例，来确定合适的移动速度以确保游戏性。

c.偏差对测试用例有效性影响的评估，test\_case\_003中的数值设定的更为合理。

# 5测试记录

a.2020.5.21 游戏原型测试

b. 2020.6.2 第一部分测试

# 6评价

6.1能力。在各个方面上能力都有所欠缺，且经验不足。没有办法比较条理地安排开发与测试工作。

6.2缺陷和限制。使用2d-game-kit上的方法比较僵硬，由于疫情原因，组员也没有办法面对面交流，工作效率比较低下。

6.3建议。下次在分工前应该对项目主体有个大致的清晰了解。主程序员模式的团队模式反而会效率高一些。

6.4结论。吸取经验教训，争取应用到以后的学习生活工作当中。

# 7测试活动总结

总结主要的测试活动和事件。总结资源消耗，如：

7.1人力消耗，由开发人员自发组织。

7.2物质资源消耗，个人电脑耗电。

-------------------------------------------------------------------------------------------------------------------------------

# 5缺陷与处理

以下是测试过程中的发现的问题与其对应的处理：玩家可以向地图的左边走，从而跳出地图外不停下坠，并卡在此状态无法结束，解决方法是添加了一些砖块形成一个高台，使得玩家无法跳跃到高台上，从而无法到达地图的另一边。此外，游戏关卡部分的探照灯速度较快，使得玩家需要反复尝试才能通过，为了体验着想我们降低了其速度数值。另外，关于在路灯间跳跃以规避探照灯的设计，由于路灯间间隔较短，在跳跃时还需要控制不让角色跳过头从而掉下路灯，与本身设计这个容易通过的隐藏通道的初衷不一致，故增大间距，使得可以直接跳到下一个路灯上。

# 6经验与教训

在项目开发的前期分工上一定在充分了解项目之后再做好对应安排，盲目的分工会导致项目开发进行效率的低下。在开发项目中与其盲目地追求所谓扁平管理从而采用“忘我”的团队模式，不如脚踏实地地采用主程序员模式，效率高的同时可以保证团队前进方向的一致。