

游戏详细设计文档

——萤火之光



团队：BlackYard



组员：乌若尧

陈旭韩

胥玲

张思语

目录

1. 游戏总体概述	6
2. 游戏设计思路	7
2.1 游戏简介	7
2.1.1 游戏类型	7
2.1.2 游戏故事摘要.....	7
2.1.3 核心玩法	7
2.1.4 平台和目标人群.....	8
2.2 游戏内容	8
2.2.1 游戏关键词.....	8
2.2.2 游戏特点	8
2.3 市场分析	9
3. 游戏故事梗概	11
3.1 故事设定	11
3.1.1 场景设定	11
3.1.2 情节设定	11
3.1.3 背景故事	12
3.2 角色设定	13
3.2.1 主要角色	13
3.2.2 反派角色	13
3.3 对话剧本	13
4. 游戏玩法设计	18

4.1 游戏规则	18
4.2 玩家能力	18
4.2.1 萤火虫能力.....	18
4.2.2 小女孩能力.....	23
4.2.3 收集要素	23
4.3 游戏世界行为规则.....	23
5. 游戏元素	25
5.1 角色设计	25
5.2 物体/机关设计	26
5.3 收集要素	35
6. 交互设计	37
6.1 PC 版操作规则	37
6.2 手机版操作规则.....	37
6.3 人机交互	37
7. 界面设计	38
7.1 界面流程图	38
7.2 UI 图例	38
7.3 开始界面设计	42
7.4 游戏界面	42
7.5 暂停界面	43
7.6 死亡界面	43
8. 游戏进程和关卡设计.....	44

8.1 第一关地图以及机关示意	44
8.2 第二关地图以及机关示意	45
9. 程序设计	46
9.1 程序总体设计	46
9.2 移动萤火虫(KeyboardControl.cs)	46
9.3 小女孩移动 (MoveControl.cs)	47
9.4 创造光球 (Apart.cs)	49
9.5 改变光照范围 (light.cs)	49
9.6 延时销毁 (DelayDestroy.cs)	50
9.7 重生 (reborn.cs)	51
9.8 机关：生成火墙 (FireWall.cs)	52
9.9 火墙自动销毁 (firetime.cs)	52
9.10 死亡脚本 (death.cs)	53
9.11 机关：普通开关 (GravityTrigger.cs)	54
9.12 机关：第一关最终开关 (FinalTrigger.cs)	55
9.13 机关：光控开关 (Lightcontrol.cs)	57
10. 开发路径规划	60
10.1 UI 素材和界面设计图制作	60
10.2 原画绘制、扫描、上色	60
10.3 模型修改、动画制作	60
10.4 音效处理	63
10.5 游戏制作和分工	63

11. 游戏系统测试	64
------------------	----

1. 游戏总体概述

游戏名称	萤火之光
小组名称	BlackYard
小组成员	乌若尧 张思语 陈旭韩 胥玲
核心概念	<p>一个小女孩在森林深处醒来，周围的一切都是漆黑的，唯一能看见的是一只发光的萤火虫。玩家将控制萤火虫，引领小女孩走出迷宫，并探索发生在女孩身上的故事。本小组策划的游戏是一款童话风的益智解谜类游戏。</p>
工作流程	<p>游戏策划讨论与玩法讨论</p> <p>素材收集与整理</p> <p>原画与界面设计</p> <p>场景搭建</p> <p>Unity 脚本编写</p>
风险管理	<ol style="list-style-type: none">1. 脚本代码生疏导致进度缓慢2. 道具和解谜元素如何设计得巧妙而且容易操作3. 部分道具和解谜元素无法实现4. 如何在没有文字叙述的情况下让剧情易于理解
预期目标	<ol style="list-style-type: none">1. 完成两个关卡的搭建2. 游戏整体测试没有大的 BUG3. 界面和 UI 设计整体效果美观大方

2. 游戏设计思路

2.1 游戏简介

2.1.1 游戏类型

《萤火之光》是一款 3D、第三人称视角、解谜类冒险游戏。

2.1.2 游戏故事摘要

游戏主要讲述了这样一个故事：小女孩醒来时发现自己身处于森林的深处，周围漆黑一片什么都看不见，只发现了发着微弱的光亮的萤火虫“点点”。她不敢到黑暗的地方去，只能在点点的光的照亮的地方活动。为了回到村子里，小女孩和萤火虫“点点”一起踏上了回家的旅途。但回到村子里的小女孩才发现自己已经在森林深处被野兽杀害，现在只是一个灵魂。村里的人为了小女孩能够安息，为她立了一块墓碑，并在墓碑前点亮了一支蜡烛。这支蜡烛感受到了村里人的思念，变化成了一只萤火虫，来到森林深处，引导小女孩的灵魂回到村里并最终安息。

2.1.3 核心玩法

本游戏最核心的玩法为鼠标操作萤火虫移动，萤火虫可以发光，但除了萤火虫发光照亮的部分以外，其他部分全部都是一片黑暗，而小女孩必须被光照到，否则会被黑暗吞噬，导致 **Game Over**。而小女孩会以一定速度自动跟着萤火虫移动。

2.1.4 平台和目标人群

游戏平台为 PC 端和手机移动端双平台。

玩家的目标人群为六岁以上的喜欢冒险类青少年人、成年人。

2.2 游戏内容

2.2.1 游戏关键词

童话风格、跨平台、玩法独特、剧情精彩

2.2.2 游戏特点

- 画面风格：**开场动画和标题画面会决定一个游戏给玩家的第一印象。我们的标题画面和 UI 界面都追求给玩家带来一种阴暗但又很温馨的感觉。我们的游戏 logo 和标题画面都很精致和细腻，玩家第一眼会被这种画面风格而吸引进而决定尝试。
- 故事情节：**游戏的整个故事安排简单但并不单一，不需要玩家花大量的时间理解，但游戏进程到最后，则会有恍然大悟的感觉。小女孩追逐萤火虫，萤火虫为小女孩照亮前面的路，这种设定，有童话的风格，带给玩家一片单纯和清新，但在结尾出人意料的结局则会给玩家带来震撼和治愈。
- 玩法设计：**游戏的玩法独特有新意。冒险类游戏有很多，但与《萤火之光》玩法完全相同的几乎没有。而本游戏又同时具有大部分冒险类游戏所使用的操作类挑战和部分解谜类游戏具有的解谜要素，加上独特的操作模式，游戏整个独特的玩法会成为游戏的一大亮点。

4. 游戏平台：我们小组最终选择把这款游戏做成同时登陆 PC 平台和手机平台的双平台游戏。作为手游，游戏可以有更大的用户群，不经常玩游戏的女士或其他成年人也有可能会尝试并喜欢上这款游戏。

2.3 市场分析

1. 平台分析：手机游戏成为了最近国内外市场新兴的一种类型。其增长势头大大影响了掌机和端游的发展方向。很多端游和掌机游戏甚至主机游戏开始选择登陆手机这一新兴平台。手游的主要优势是开发成本较低、用户广、受众多，所以我们的游戏选择同时开发 PC 版和手游版，迎合这一市场热潮。

2. 游戏类型分析：解谜类游戏是独立游戏里很常见的游戏类型，原因是开发难度低、不需要太高的渲染技术。冒险类游戏则是很一直热门的游戏类型。两者的结合很值得期待。

3. 同类游戏对比：同类游戏中比较成功的当属《纪念碑谷》、《时空幻境》、《地狱边境》这三款游戏。这三款游戏都不是大制作的游戏，但都取得了相当好的评价和销量。对比《纪念碑谷》和《时空幻境》，两款游戏都有独特的玩法，《纪念碑谷》的玩法是通过 3D 和 2D 之间的视觉错觉而解决关卡，《时空幻境》则是以其独特的时间轴玩法被众人称赞。两款作品说明独特的游戏玩法会给玩家带来独特的感受，进而被玩家喜欢。虽然不敢说能达到这两款作品玩法上的高度，但《萤火之光》的玩法也十分独特，也能吸引玩家。对于谜题的设计，《地

狱边境》则是一个很好的例子。一款游戏不需要精致的画面，只需要对于谜题巧妙地把握便可以吸引玩家。在童话风格的游戏里，我们参考了《光之子》和《萤火虫日记》这两款游戏。《光之子》是育碧旗下少有的一款 JRPG 游戏，讲述了少女“欧若拉”被迫远离父母，为了回到家而踏上冒险的旅途的故事。《萤火虫日记》则是一款更为小众的独立游戏，它的操作更难故事更简单，其漂亮的人设也为这款游戏增色不少。对比这两款游戏，游戏的主要风格和玩法都有相同之处，都是有关小女孩的童话风格，都是需要同时操作小女孩和搭档共同解谜。区别在于《光之子》拥有更多的关卡和剧情，这或许让游戏看起来更丰富，但这使《光之子》无法登陆手机端，毕竟手游玩家很少有大块的时间和精力来理解一款游戏，而《萤火虫日记》则因为其难度过高，是很多玩家望而却步。而我们的《萤火之光》则刚好吸收了它们的精华，去掉了它们的一些特点使游戏更加大众化。**综上所述：**《萤火之光》这款游戏既有自己的独特之处，又有成为热门游戏所必须的一些要素，故我们认为这是一款成功的设计。

3. 游戏故事梗概

3.1 故事设定

3.1.1 场景设定

场景分为三幕，对应为三个关卡。

第一幕在深林里，从小女孩醒来、与萤火虫“点点”相遇，到通过最后一个关卡，走入山洞中。

第二幕在山洞里，虽然是走进了山洞，但实际上看上去更像地牢，这里的设定是地狱的边境，如果没能通过这里，小女孩就只能变成厉鬼而无法安息。

第三幕是在村子里。村子里有着她熟悉的一切，她路过一切她曾玩耍过的地方以后，最终来到了村里人为她修的墓前。

3.1.2 情节设定

（情节设定里面没有写对话，所有的对话都在剧本里单独写出）

故事开始于小女孩在深林中醒来。醒来的小女孩发现自己身处与一片黑暗的树林之中，伸手不见五指，有关自己身世的一切都想不起来。在她身边有一只会发光的萤火虫，自称为“点点”。点点的光亮只能照亮很小的一片地方，小女孩不敢走进黑暗中，因为黑暗中充满了未知的危险，随时都有可能死掉。点点看到小女孩在害怕，决定用自己的光亮带领小女孩走出深林，找到她的归所。小女孩先穿过一片树林，在路程上点点不断安慰小女孩，和她聊天使他放松。穿过树林

以后他们进入一个山洞里面，在山洞里面看到了许多可怕的怪兽，点点和小女孩相互鼓励，共同克服各式各样的难关，小女孩也渐渐找回了自己的记忆。最后从山洞里面出来，来到了一座小村庄，穿过村庄，小女孩想起了她在这里玩耍时快乐的日子。就当她自己以为可以和父母相见，重新回到日常的生活中的时候，小女孩走到了一座墓前，发现墓前的牌子上刻着自己的名字，这时小女孩渐渐变得透明。她终于想起来自己在深林里被野兽袭击，已经死去了。这时她再也无法承受心中的失望和伤心，哭了出来。点点在这时飞到她的身前，化作她的母亲，给了她一个拥抱。最终小女孩决定接受这一切，早点安息转世，再次作为母亲的孩子生活。于是她化作灵魂沉睡在了这座墓里。而原本的萤火虫慢慢飘下，变成一支蜡烛，在墓前发光。很快村里人聚集过来给她上坟。人群聚集以后又散开，最后留下一片各式的蜡烛，照亮着小女孩前行的路，希望她能早日转世归来。

3.1.3 背景故事

本游戏所讲述的故事在一开始可能被认为是一个萤火虫带领小女孩回家的故事。但随着情节的不断深入和萤火虫和小女孩之间对话的暗示，玩家会逐渐发现故事没有那么简单。小女孩的真身因为在森林里迷路被野兽袭击了，村里的人们找到了她的尸体并安葬了她。但她的灵魂仍在森林里，徘徊在地狱的边缘。萤火虫实际上是母亲在她墓前摆的蜡烛，母亲希望小女孩安息的强烈的愿望使蜡烛变成了萤火虫，去引领小女孩从地狱的边缘离开并找到自己的真身，真正安息。

其中小女孩走过的山洞便是地狱的边境。最后村里人摆放的蜡烛，是在引导她早日归来。

3.2 角色设定

3.2.1 主要角色

1. 小女孩

游戏的主人公。醒来时发现自己身处于森林的深处，周围漆黑一片什么都看不见，只发现了发着微弱的光亮的萤火虫“点点”。十分胆小不敢到黑暗的地方去，只能在点点的光的照亮的地方活动。

2. 萤火虫“点点”

玩家主要操作的角色，解谜部分的主力。由小女孩村里的人们的在小女孩墓前摆放的蜡烛的烛光和人们的思念变化而来的萤火虫，目的是把小女孩的灵魂引导回到村子里是她能够安息。在路上负责安慰小女孩和解释一些机关、能力的用法。

3.2.2 反派角色

游戏里没有明确的角色和主角是对立的。所以并没有明确的反派角色。玩家需要对抗的是各种机关和迷宫等解谜要素。

3.3 对话剧本

【游戏开始动画】

小女孩：……（睁眼画面打开）好黑啊… 这里是哪里？（眨眼）

那是……（视野出现一个亮点）

点点：（笑声）你终于醒来了，来吧。跟我走吧，我会带你去你该去的地方。

小女孩：“我的……地方？”

（萤火虫转身，小女孩跟上，画面切换至**游戏开始菜单**）

【树林】

（非动画部分 此处对话会在游戏过程中随机产生）

点点：“你别小看我，我可是森林里最聪明的萤火虫，永远不会在森林里迷路的哟！”

点点：“好饿啊！我们要快点出去吃东西！”

小女孩：“点点，我们可以休息一下吗？我好累哦！”

点点：“亲爱的，你要再坚持一下！我也想让你休息啊。可是，等到天亮我们就出不去了！”

点点：“我们做好朋友好不好？你可以叫我点点，我就叫你小点儿喽！”

点点：“好冷啊！我们还是再走快点吧！”

点点：“点点最喜欢吃棒棒糖，小点儿，你喜欢吃什么？”

【第一关通关动画部分，树林转山洞】

点点：“我们走出了不属于你的地方，现在通过了这个山洞就是你的归属所在。”

小女孩：“我的.....归属？”（突然一个模糊的画面，是一个慈祥的女人笑着俯身看着自己，手里拿着糖，正在向自己伸来）

点点：“这里比树林还容易走丢哦，你一定要乖乖的！”

小女孩：“嗯嗯.....我会跟着你的。”

【山洞】

（非动画部分 此处对话会在游戏过程中随机产生）

点点：“这里很黑，一定要紧紧跟着我哦！”

点点：“小心脚下，踩到那些地刺，我们就回不去了！”

小女孩：“~~~~(>_<)~~~~ 我要回家！我一定会小心的！”

小女孩：“这个山洞里有怪兽吗？”

点点：“当然有啊！所以我们要更小心了！”

小女孩：“点点，点点，你走慢一点！我快要跟不上你了！”

点点：“快一点！马上就要天亮了！只要天亮，我们就出不去了。”

点点：“这个是火墙，不要怕！过来就好了！”

小女孩：“这里好黑啊！我们可以把这里照亮一点吗？”

点点：“可以，只要把我的小灯笼调亮就好了！”

【村庄】

小女孩：“这里……是……”

点点：“我们就快要到了 …… 快要 …… 到了 ……”

小女孩：“点点，你怎么了？我们不是终于要到家了么？”

点点：“是啊 …… 我们就要到家了 ……”

（这里小女孩每走过一个地方就会有一幅反映她曾在这里玩耍过的图片。）

【通关动画部分】

（小女孩走到了一座小小的土包前面，前面有一个小小的木牌，木牌上的字已看不清，木牌前的地方还放着一个棒棒糖，被糖纸裹得紧紧的，以至于都没有蚂蚁过来纠缠。她仿佛一下记起了什么。安静的飘在空中，扇动着翅膀）

小女孩“萤火虫.....进了山洞.....进了树林.....”（*看向自己的下半身，自己身体半透明化*）

（*场景不断闪现，分别展示小女孩离开村子、在森林里迷路、被野兽杀掉的画面*）

点点：“对不起，亲爱的，我的任务是把你带到这里来，虽然...虽然你会消失...我...我...”

小女孩：“点点...没关系的，点点，我不怕。通过这一路的经历，我懂了很多事。没关系的，我会坚强，我会安息的。这样我才能转世，才能再次回到村里来。所以，所以，谢谢你，点点。”

点点：“我相信你，你一定会回来的。这里是爱你的村民们为你建的家，他们希望我能够把你带出森林。希望你能够安息，希望你能回来。所以，你永远都不会是一个人的！我也会，和他们一起在这里等你！”

小女孩：“爱我的人.....”（*再浮现之前画的母亲画面*）“妈妈.....村长伯伯.....还有叔叔阿姨，哥哥姐姐，谢谢你们！”（*小女孩化作一片荧光，然后消失*）

（*镜头正对土包和木牌，点点缓缓地飘落，最后化成一支蜡烛。村里的人们聚集了过来，此处只有大家的背影。人们越聚越多，又各自走开，最后只剩小女孩的母亲，周围摆满了刚才村民们摆好的各式的蜡烛。周围逐渐变白，最终只剩萤火虫化作的蜡烛。出现通关画面。*）

4. 游戏玩法设计

4.1 游戏规则

游戏的**核心玩法**是萤火虫存在的地方会照亮一个特定的范围，小女孩只能在这个范围里面活动，一旦萤火虫移动速度过快使小女孩完全处于黑暗中时，小女孩死亡，游戏结束。游戏的**根本目的**是保护小女孩，并把小女孩带回村庄。小女孩的移动是自动的，跟随萤火虫当前所在位置移动。小女孩所处场景是一个**迷宫**，里面有各种**机关**。玩家需要在保证小女孩不死亡的前提下，破解机关，通过关卡，最终达成通关。同时在每个关卡的迷宫里面还有特定的收集品，收集要素并不影响游戏的整体流程和故事结局，仅作为游戏的扩展元素存在。

4.2 玩家能力

4.2.1 萤火虫能力

（萤火虫最基本的能力是照亮环境，保护小女孩不被黑暗吞噬。）

以下是基于基本能力发展出来的其他技能，其开启时间点并不相同。



图 4.1

1. 控制小女孩移动

操作方式：键盘 WASD 控制萤火虫移动，小女孩会跟随萤火虫。

解锁时间：游戏开始

玩法说明：小女孩会跟着萤火虫移动

涉及到的机关：地刺、火墙

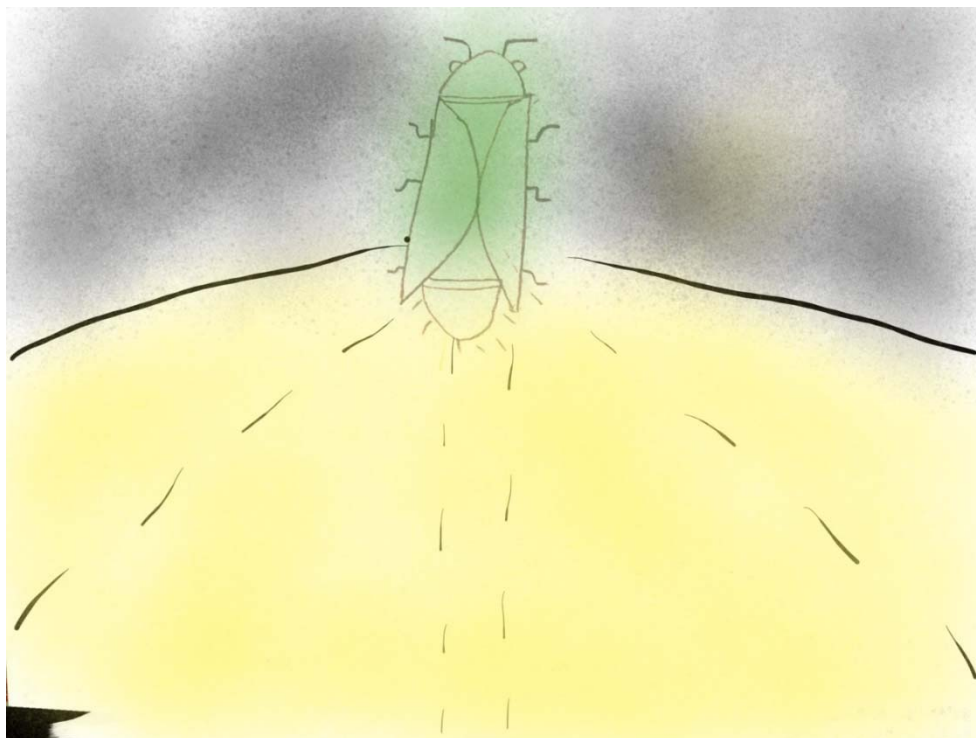


图 4.2



图 4.3

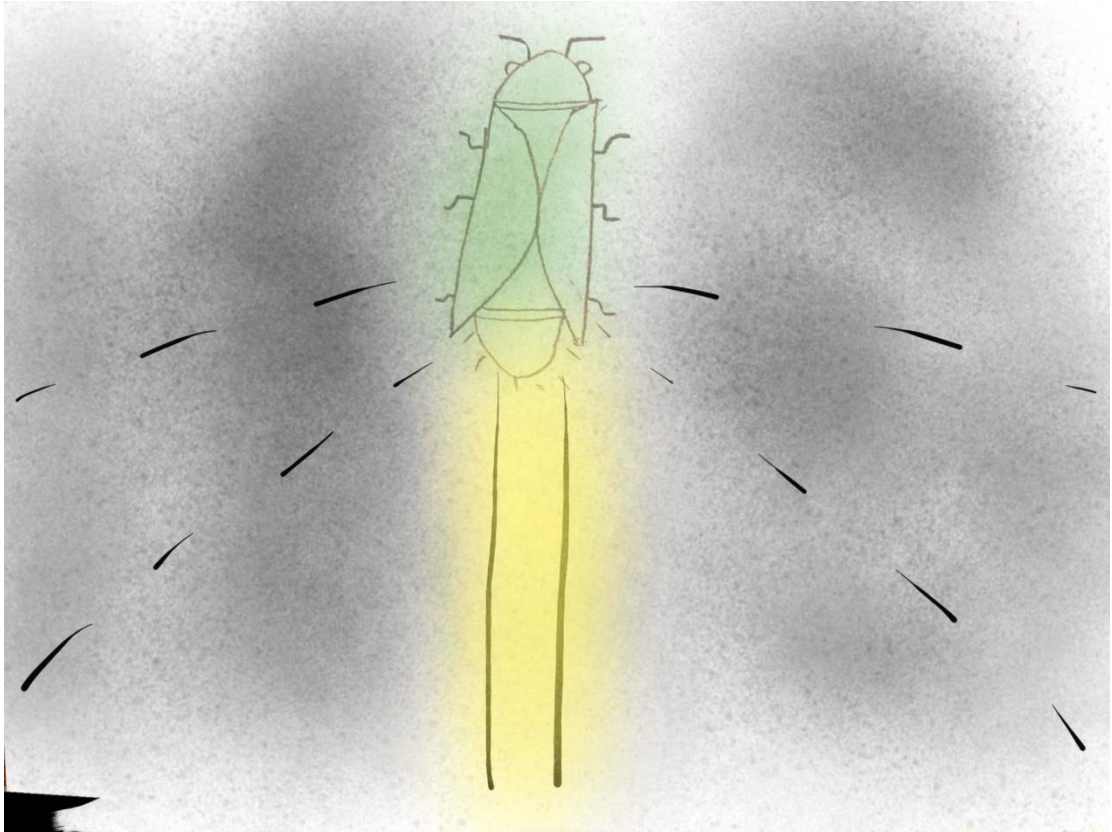


图 4.4

2. 改变光照范围

操作方式：鼠标点击特定按钮

解锁时间：第二关初

玩法说明：按钮可以改变萤火虫的光照范围，光照范围越大，光强越弱。最小范围的光照不够容纳小女孩，但其光强可以点燃物品。

涉及到的机关：枯木、黑影



图 4.5

3. 创造光球

操作方式：鼠标点击特定按钮

解锁时间：第一关初

玩法说明：萤火虫可以创造一个短暂的光球，小女孩可以在一定时间内停留在光球周围，而萤火虫可以独立行动

涉及到的机关：枯木、光控开关

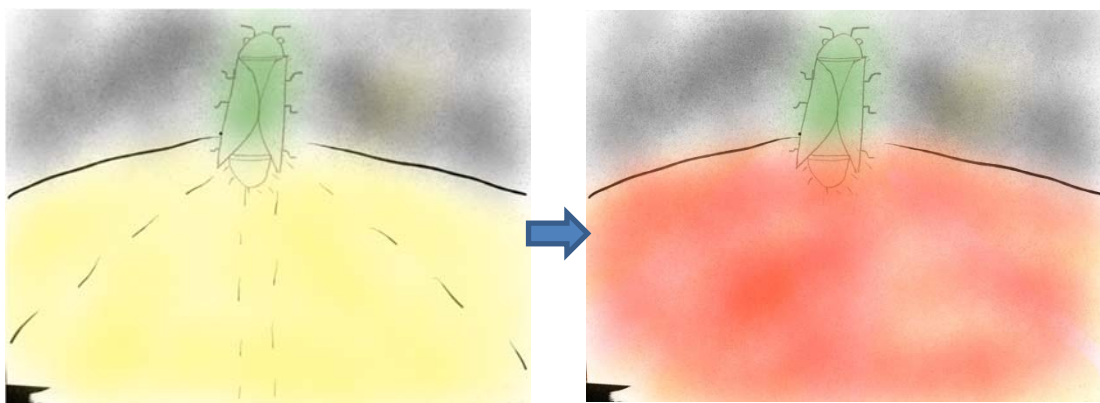


图 4.6

4. 变色发光

操作方式：鼠标点击特定按钮

解锁时间：第三关初

玩法说明：萤火虫可以改变自身发光的颜色

4.2.2 小女孩能力

跟随萤火虫和触发特定的机关。

（有关小女孩触发的机关参考“物体/机关设计”）

4.2.3 收集要素

游戏里在除各种机关以外，还会在迷宫的某些地方出现特定的收集物品。每一关有 5 个，收集品的收集程度会在每一关的开始提示给玩家。收集要素并不影响游戏的整体流程和故事结局，仅作为游戏的扩展元素存在。可以成为游戏多周目的目的之一。

4.3 游戏世界行为规则

规则说明以“4.1 游戏规则”里的介绍为主。游戏世界地形不可

被改变，搭建迷宫的元素不可被穿过，小女孩只能在规定的地方里面活动。萤火虫的技能只对特定机关有效，不能对环境产生影响。游戏世界的物理规则基本符合现实。对于特殊机关的设定请参考“5.2 物体/机关设计”这里不予赘述。

5. 游戏元素

5.1 角色设计



图 5.1

小女孩：

姓名：不详

年龄：12 岁

身高：134cm

性格：胆小、天真、坚强

说明：游戏的主人公。醒来时发现自己身处于森林的深处，周围漆黑一片什么都看不见，只发现了发着微弱的光亮的萤火虫“点点”。十分胆小不敢到黑暗的地方去，只能在点点的光的照亮的地方活动。性格天真可爱。虽然很胆小但在关键的时候很坚强。最后发现了自己已

经死去的事实。虽然伤心但最后还是坚强地接受了一切。愿意最后安息，并转世重新回到村子里来。

萤火虫：

（形象不需要特殊设定，在游戏里只是一团光的存在）

昵称：点点

年龄：12 小时

身长：3cm

性格：开朗、勇敢

说明：玩家主要操作的角色，解谜部分的主力。由小女孩村里的人们的在小女孩墓前摆放的蜡烛的烛光和人们的思念变化而来的萤火虫，目的是把小女孩的灵魂引导回到村子里是她能够安息。在路上负责安慰小女孩和解释一些机关、能力的用法。说话风趣幽默，每一句话都为了让小女孩感到安全，消除紧张感。但谈吐之间又有一种深意。知道自己的使命只是带小女孩回到村里安息，但对于小女孩的经历在心中有着一种同情，随着交往的深入这种同情变成了不舍。知道自己会让小女孩最终消失但还是坚持着完成自己的使命。

5.2 物体/机关设计

（这里展示的是关卡里会多次出现得通用机关，有的关卡会有一关里所特有的机关。）

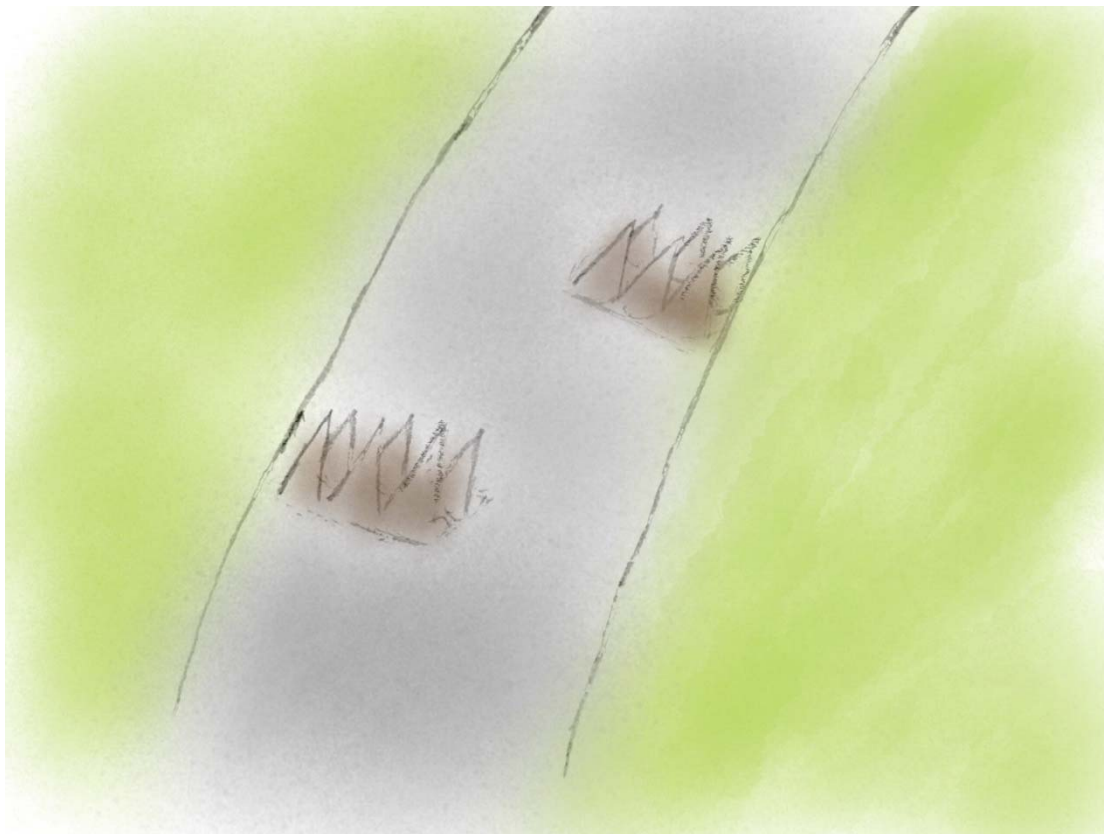


图 5.2

1. 名称：地刺

描述：出现在地表的尖锐的刺，会刺伤小女孩

出现地点：较长走廊的地面上

触发条件：小女孩触碰到

触发结果：小女孩死亡

玩家挑战：由于玩家不能直接操作小女孩故带来操作上的挑战

难度评价：普通

特别说明：此处指代一类机关，这类机关并不局限于地刺这种表现形式



图 5.3

2. 名称：门

描述：厚重的石门阻挡了小女孩前进的道路

出现地点：必经道路上的任意位置

触发条件：触发特定的开关

触发结果：门消失，道路可以通过

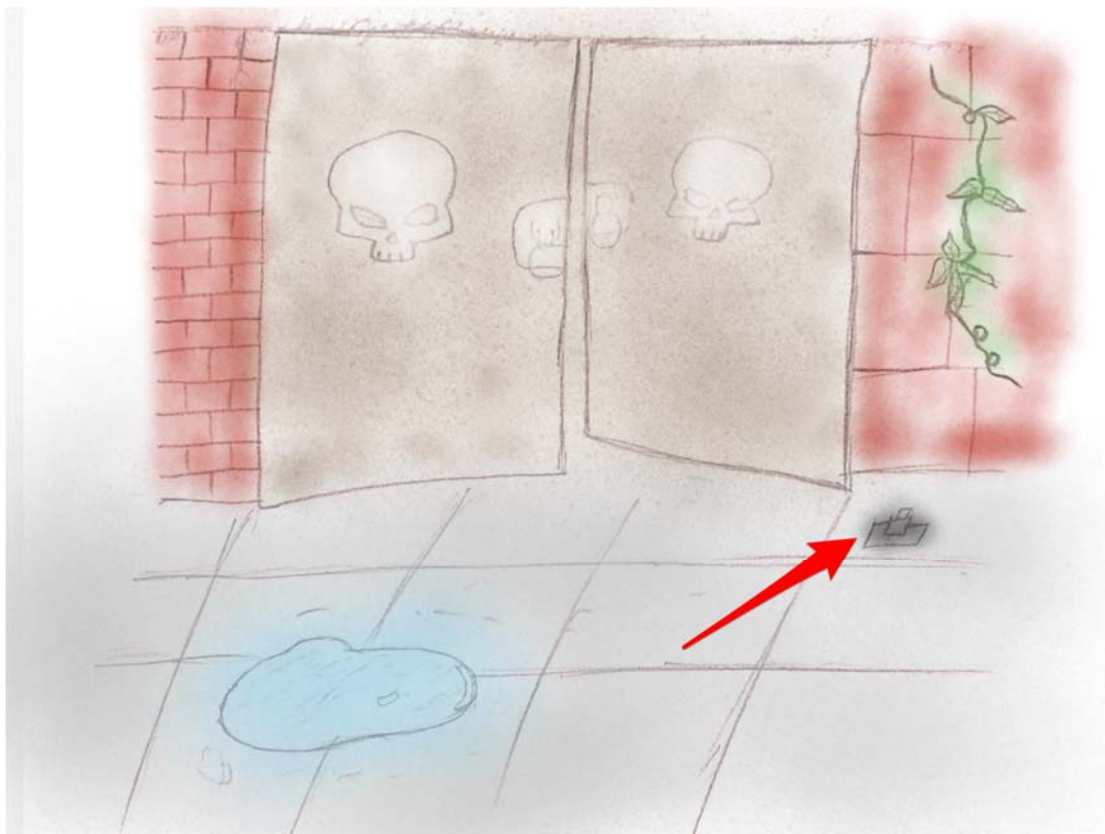


图 5.4

3. 名称：普通开关

描述：地面上的凸起，看起来可以踩下去

出现地点：要通过的门之前的任意位置

触发条件：小女孩触碰到

触发结果：开启特定的门

玩家挑战：需要认真寻找，同时普通开关还可以和其他开关共同触发

开门事件，需要玩家认真观察

难度评价：简单



图 5.5

4. 名称：光控开关

描述：地面上的宝石闪烁着别样的光辉，照亮它似乎可以引发共鸣

出现地点：一般在要通过的门后面的任意位置

触发条件：萤火虫处于开关正上方

触发结果：开启特定的门

玩家挑战：需要认真寻找，同时光控开关还可以和其他开关共同触发

开门事件，需要玩家认真观察

难度评价：普通

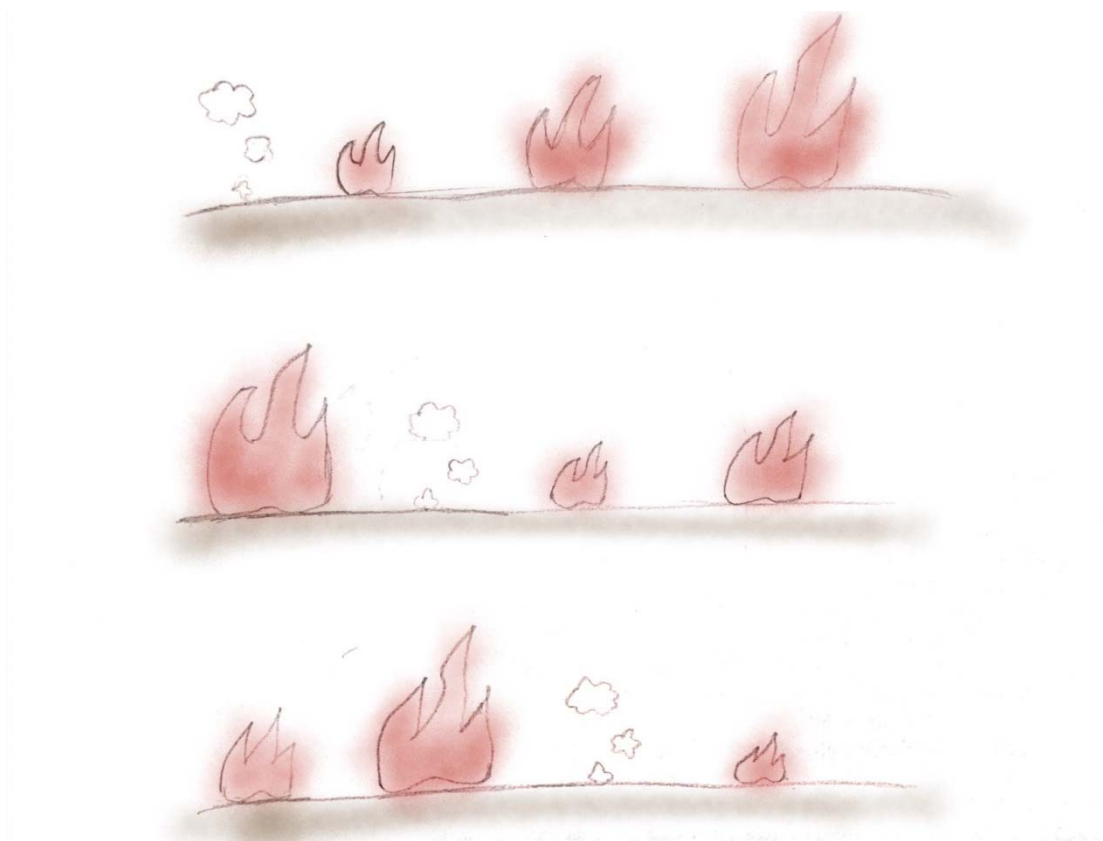


图 5.6

5. 名称：火墙

描述：熊熊烈火燃烧着，仿佛要燃尽周围的一切

出现地点：任意走廊的地面上，同时出现多个，会间接消失

出现个数：3-5 个/次

触发条件：小女孩触碰到

触发结果：小女孩死亡

玩家挑战：由于玩家不能直接操作小女孩故带来操作上的挑战，由于间接消失，所以要抓准时机快速通过

难度评价：困难

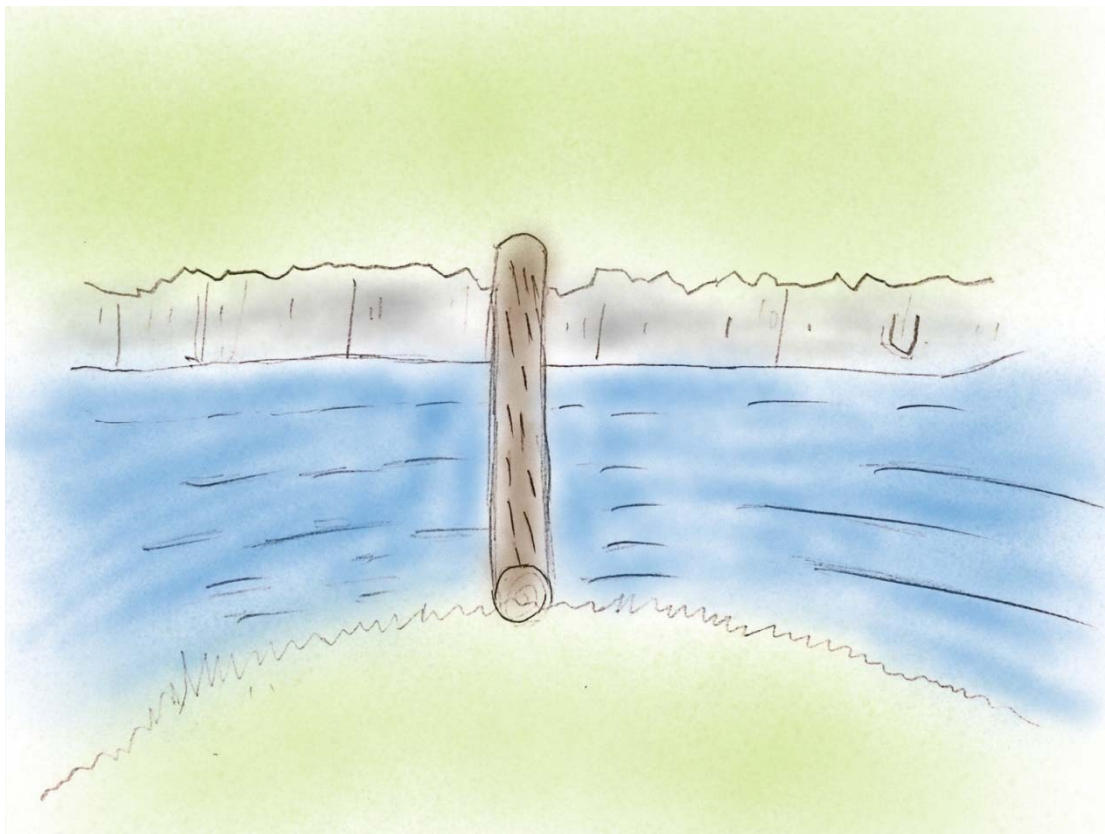


图 5.7

6. 名称：独木桥

描述：狭窄的通道刚好可以让小女孩走在上面，但一不小心就有可能掉入万丈深渊

出现地点：较长走廊的地面上

触发条件：小女孩离开桥面

触发结果：小女孩死亡

玩家挑战：由于玩家不能直接操作小女孩故带来操作上的挑战

难度评价：困难

特别说明：此处指代一类机关，这类机关并不局限于一种表现形式



图 5.8

7. 名称：枯木

描述：倒在道路中央的枯木，似乎可以点燃

出现地点：任意位置

触发条件：萤火虫光照范围最小

触发结果：枯木点燃

玩家挑战：需要将小女孩留在其他位置，萤火虫独立完成

难度评价：简单

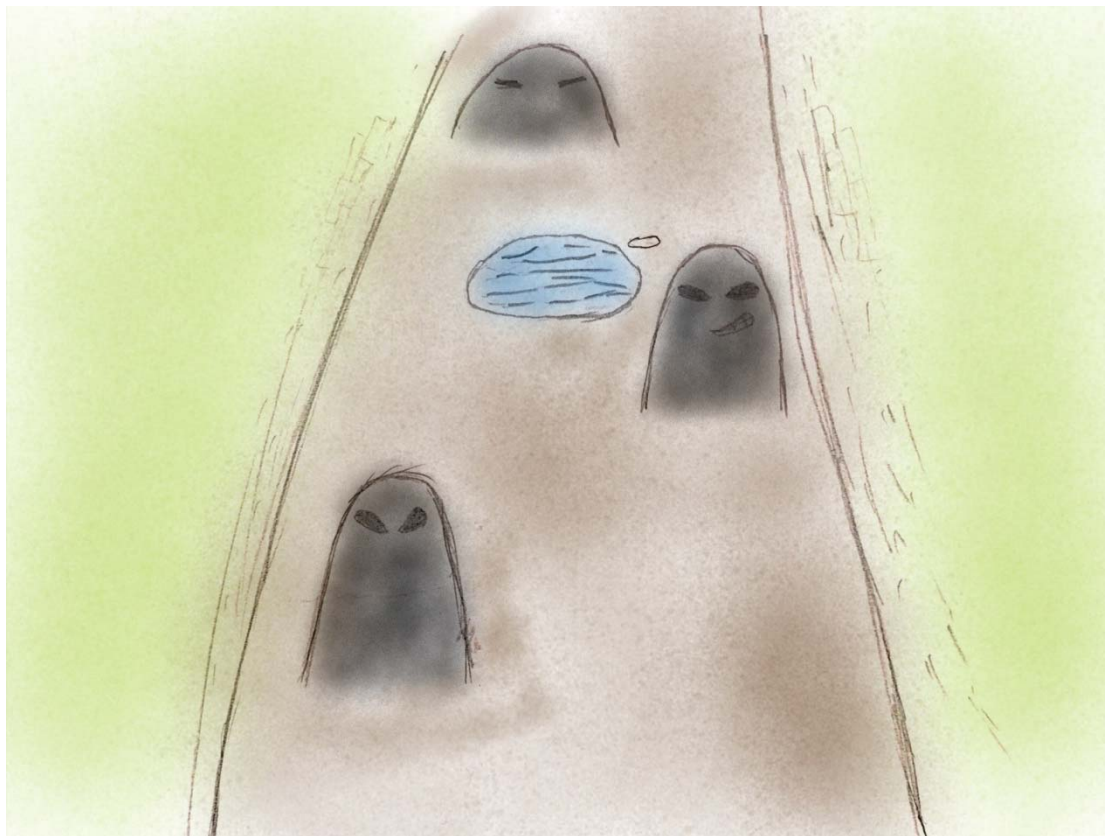


图 5.9

8. 名称：黑影

描述：隐藏在黑暗中的黑影，被光照到的话可能会被惊吓而扑过来

出现地点：任意位置

触发条件：被萤火虫发的光照到

触发结果：小女孩被黑暗吞噬

玩家挑战：黑影的位置很特殊需要玩家调整萤火虫发光范围才能带领小女孩躲过去

难度评价：普通

5.3 收集要素

描述：游戏里在除各种机关以外，还会在迷宫的某些地方出现特定的收集物品。每一关有 5 个，收集品的收集程度会在每一关的开始提示给玩家。收集要素并不影响游戏的整体流程和故事结局，仅作为游戏的扩展元素存在。可以成为游戏多周目的目的之一。

收集品设定为小女孩生前所喜欢的各类物品。



图 5.10

收集品 1：小熊



图 5.11

收集品 2：棒棒糖

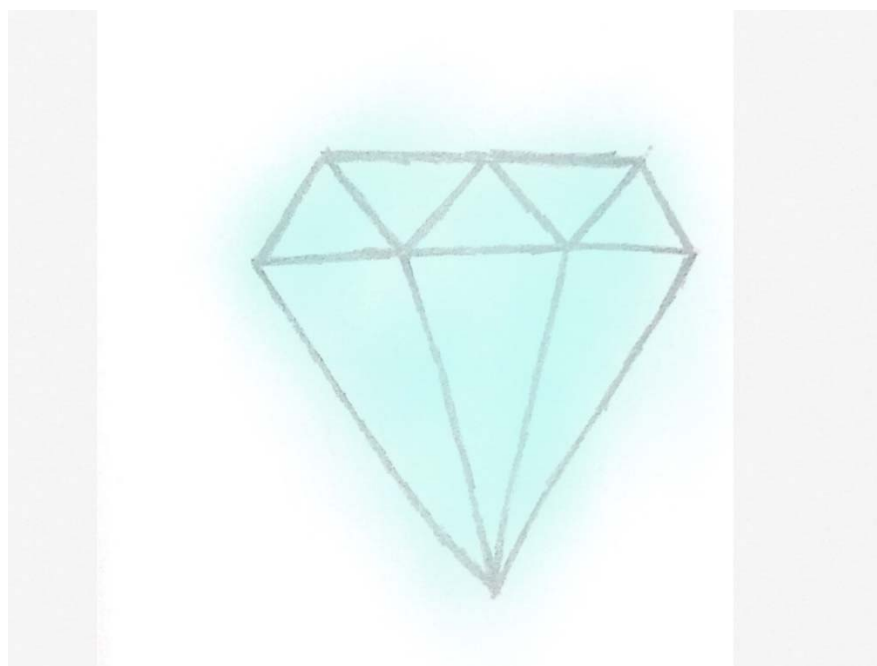


图 5.12

收集品 3：发光的宝石

6. 交互设计

6.1 PC 版操作规则

萤火虫移动：WASD 分别控制上下左右

技能使用：鼠标点击按钮

暂停、静音、退出：鼠标点击特定按钮

（按钮位置参照界面设计）

6.2 手机版操作规则

萤火虫移动：手在触摸屏的任意位置向某个方向推动一个很小的距离后不动，萤火虫即可向那个方向持续移动直至手抬起。

技能使用：手指点击特定按钮。

暂停、静音、退出：鼠标点击特定按钮

（按钮位置参照界面设计）

6.3 人机交互

（参照界面设计）

7. 界面设计

7.1 界面跳转关系图

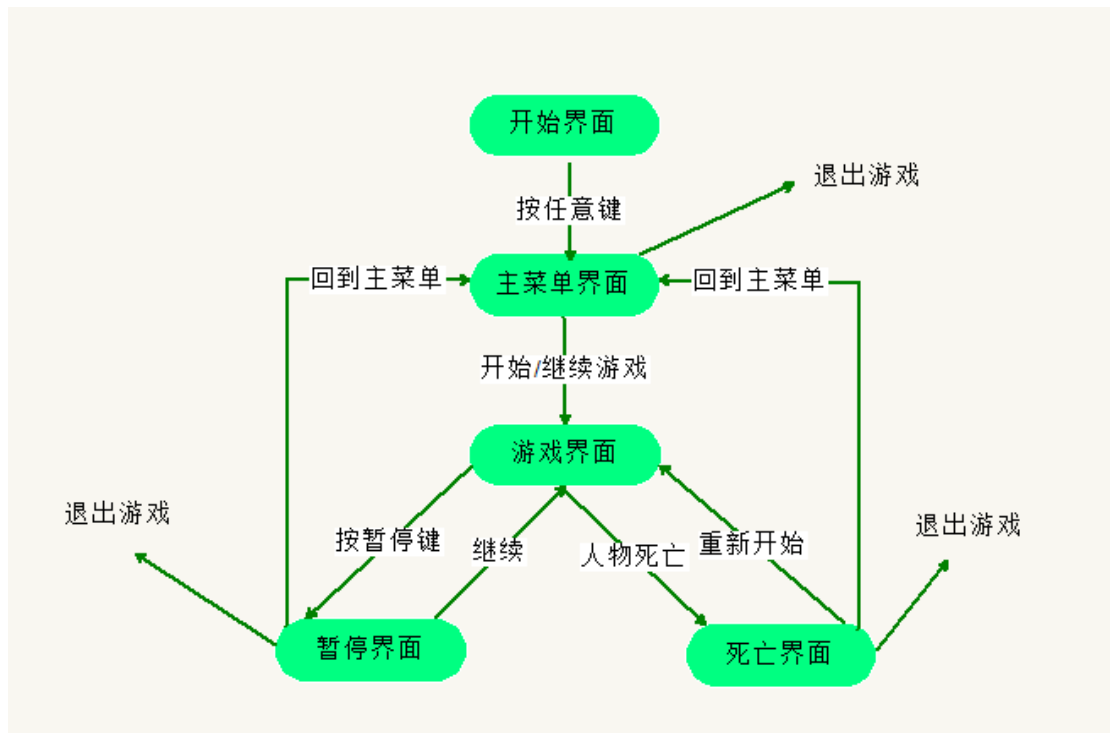


图 7.1

7.2 UI 图例

A 技能键:



图 7.2 变色发光键



图 7.3 创造光球



图 7.4 改变光照范围

B 其他功能按钮：



图 7.5 暂停键



图 7.6 静音键



图 7.7 继续游戏键



图 7.8 重新开始



图 7.9 退出游戏



图 7.10 返回主菜单按钮

7.3 开始界面设计



图 7.11

7.4 游戏界面

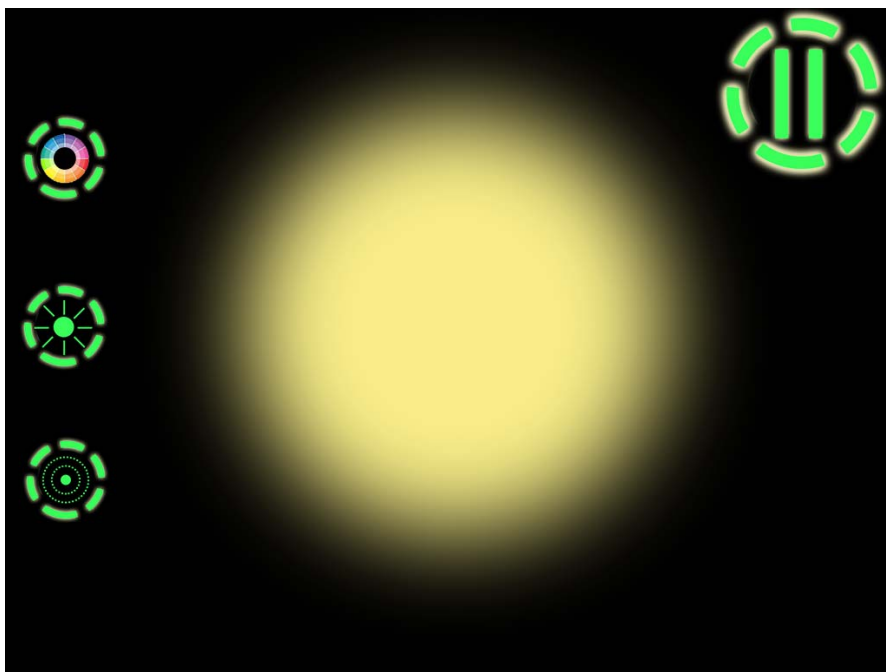


图 7.12

7.5 暂停界面



图 7.13

7.6 死亡界面



图 7.14

8. 游戏进程和关卡设计

8.1 第一关地图以及机关示意

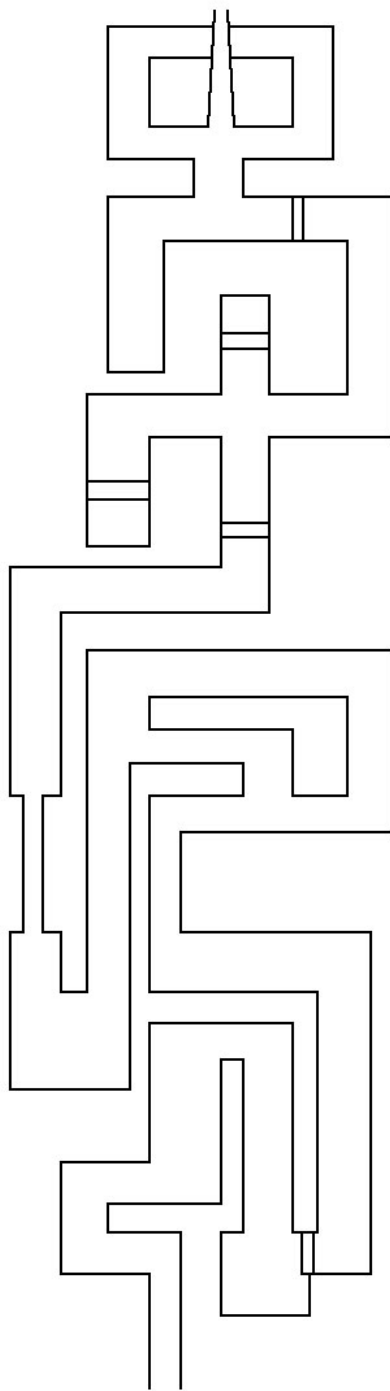


图 8.1 第一关地图

8.2 第二关地图以及机关示意

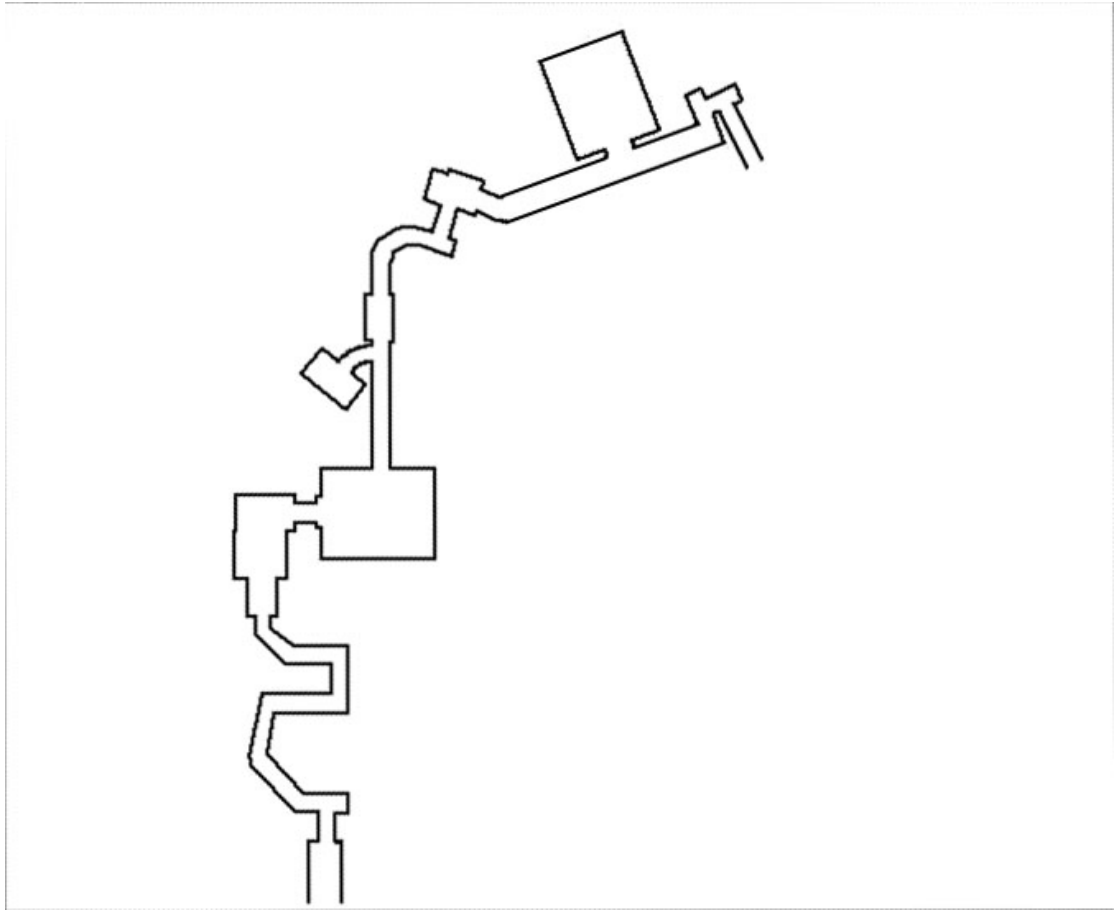


图 8.2 第二关地图

9. 程序设计

9.1 程序总体设计

KeyboardControl.cs 控制萤火虫的移动，键盘 W/A/S/D

MoveControl.cs 控制小女孩面向萤火虫跟随移动

Apart.cs 控制萤火虫技能“创造光球”

light.cs 控制萤火虫技能“改变光照范围”

DelayDestroy.cs 延时销毁

Reborn.cs 小女孩死亡后重生脚本

FireWall.cs 生成防火墙脚本

firetime.cs 防火墙自动销毁

death.cs 小女孩的死亡控制脚本

GravityTrigger.cs 普通开关的实现脚本

FinalTrigger.cs 第一关最终

Lightcontrol.cs 光控开关

9.2 移动萤火虫(KeyboardControl.cs)

```
using UnityEngine;
using System.Collections;

public class KeyboardControl : MonoBehaviour {
    public float speed = 1;
    // Use this for initialization
    void Start () {
    }

    // Update is called once per frame
```

```
void Update () {
    if(Input.GetKey (KeyCode.W))
    {
        transform.position += new Vector3(0f, 0f, speed);
    }
    else if(Input.GetKey (KeyCode.S))
    {
        transform.position -= new Vector3(0f, 0f, speed);
    }
    if(Input.GetKey (KeyCode.D))
    {
        transform.position += new Vector3(speed, 0f, 0f);
    }
    else if(Input.GetKey (KeyCode.A))
    {
        transform.position -= new Vector3(speed, 0f, 0f);
    }
}
}
```

9.3 小女孩移动 (MoveControl.cs)

```
using UnityEngine;
using System.Collections;

public class MoveControl : MonoBehaviour {

    public static float hight = 0.05f;

    public float moveSpeed = 10f;//相机移动速度
    public static bool play = true;

    public Transform player;//定义一个人物的 Transform
    public Transform girl;
    public Transform target;
    public Transform c;

    Plane playerPlane;
    Vector3 targetposition;
    float dist;//定义一个 float 变量(该变量会在下面的代码里用到)
    Vector3 cursorScreenPosition;
    Ray ray;

    public float range;
```

```
// Use this for initialization
void Start () {
    playerPlane = new Plane(player.up, player.position); //在人物所在位置, 以人物的 Vector3.up 为法线创建一个平面
    player.rotation = Quaternion.identity;
    c.rotation = Quaternion.identity;
}

// Update is called once per frame
void Update () {
    transform.position += Vector3.right * (target.position.x - transform.position.x) * Time.deltaTime * moveSpeed; //当鼠标平移时, 相机随之平移(或右移或左移)
    transform.position += Vector3.forward * (target.position.z - transform.position.z) * Time.deltaTime * moveSpeed; //当鼠标纵移时, 相机随之纵移(或上移或下移)
    targetposition = player.TransformPoint(new Vector3(0f, height, (float)(-0.5*height)));
    transform.position = targetposition; //相机的目标位置, 这两句代码的作用是让人物一直处于相机的视野下
    if(play)
    {
        PlayerMove();
    }
    player.rotation = new Quaternion(0f, 0f, 0f, 0f);
    c.rotation = Quaternion.identity;
}

void PlayerMove()
{
    // print ("bbbb");
    // cursorScreenPosition = Input.mousePosition; //鼠标在屏幕上的位置
    // ray = Camera.main.ScreenPointToRay(cursorScreenPosition); //在鼠标所在的屏幕位置发出一条射线(暂名该射线为 x 射线)

    // bool b = playerPlane.Raycast(ray, out dist); //人物所在平面与 x 射线相交(这句代码的作用是得到 x 射线与人物所在平面相交时, x 射线射出多远——即得到 dist 的值)
    // targetposition = ray.GetPoint(dist); //在 x 射线这条射线上, 距离起点 dist 处的点(该点即为鼠标在世界空间里的坐标)

    if(Mathf.Abs (target.position.x - player.position.x) > range ||
        Mathf.Abs (target.position.z - player.position.z) > range)
```



```
{
// print ("aaaa");
targetposition = target.position - new Vector3(0f, 20f, 0f);
c.position = Vector3.MoveTowards(c.position, targetposition, Time.deltaTime*moveSpeed); //人物坐标移动到鼠标所在世界空间的坐标
girl.transform.LookAt (targetposition);
}
}
}
```

9.4 创造光球 (Apart.cs)

```
using UnityEngine;
using System.Collections;

public class Apart : MonoBehaviour {
public static bool play = true;
public GameObject prefab;
public Transform c;
// Update is called once per frame
void Update () {
if(play)
{
if(Input.GetKeyDown (KeyCode.Space))
{
play = false;
MoveControl.hight = 0.07f;
c.position = c.position;
Instantiate (prefab, transform.position, transform.rotation);
MoveControl.play = false;
Death.play = false;
}
}
}
}
```

9.5 改变光照范围 (light.cs)

```
using UnityEngine;
using System.Collections;

public class Light : MonoBehaviour {
public static bool light = false;
```

```
// Update is called once per frame
void Update () {
    if(Input.GetKeyDown (KeyCode.F))
    {
        light = !light;
        ChangeLight (light);
    }

}

void ChangeLight(bool light)
{
    if(light)
    {
        Death.r = -1f;
        gameObject.light.spotAngle = 30f;
        gameObject.light.range = 50f;
    }
    else
    {
        Death.r = 225f;
        gameObject.light.spotAngle = 100f;
        gameObject.light.range = 30f;
    }
}
}
```

9.6 延时销毁（DelayDestroy.cs）

```
using UnityEngine;
using System.Collections;

public class DelayDestroy : MonoBehaviour {

    // Use this for initialization
    void Start () {
        StartCoroutine (Fun());
    }

    public IEnumerator Fun()
    {
        yield return new WaitForSeconds(10f);
        MoveControl.play = true;
        Death.play = true;
    }
}
```

```
Apart.play = true;
MoveControl.hight = 0.05f;
Destroy (this.gameObject);
}
}
```

9.7 重生 (reborn.cs)

```
using UnityEngine;
using System.Collections;

public class Reborn : MonoBehaviour{

public Transform reborn1;
public Transform reborn2;
public Transform reborn3;
public static float position_x,position_y,position_z;
void OnTriggerEnter(Collider c){
    if (c.transform.gameObject==reborn3.transform.gameObject)
    {
        position_x=c.transform.position.x;
        position_y=c.transform.position.y;
        position_z=c.transform.position.z;
    }
    else if (c.transform.gameObject==reborn2.transform.gameObject)
    {
        position_x=c.transform.position.x;
        position_y=c.transform.position.y;
        position_z=c.transform.position.z;
    }
    else if (c.transform.gameObject==reborn1.transform.gameObject)
    {
        position_x=c.transform.position.x;
        position_y=c.transform.position.y;
        position_z=c.transform.position.z;
    }
}
}
```

9.8 机关：生成防火墙（FireWall.cs）

```
using UnityEngine;
using System.Collections;

public class FireWall : MonoBehaviour {
    public Transform fire;
    int i = 0;
    float time = 0f;
    // Update is called once per frame
    void Update () {

        if (Time.time - time >= 10f)
            i = 0;
        if (i < 25)
        {
            //Debug.Log (i);
            if (i == 0) time = Time.time;
            Instantiate(fire, new Vector3(this.transform.position.x + i * 1.0f,
this.transform.position.y, this.transform.position.z),
Quaternion.identity);
            i++;
        }
    }
}
```

9.9 防火墙自动销毁（firetime.cs）

```
using UnityEngine;
using System.Collections;

public class firetime : MonoBehaviour {

    public float lifetime = 4f;
    float time = Time.time;
    // Use this for initialization
    void Start () {
        //float time = Time.time;
        //Debug.Log (time);
        //Debug.Log (Time.time);
    }

    // Update is called once per frame
```

```
void Update () {  
  
    if (Time.time>time+lifetime){  
        /* Debug.Log ("time");  
        Debug.Log (time);  
        Debug.Log (Time.time);*/  
        Destroy (this.gameObject);  
    }  
}  
}
```

9.10 死亡脚本（death.cs）

```
using UnityEngine;  
using System.Collections;  
  
public class Death : MonoBehaviour {  
    public Transform sphere;  
    public Transform player;  
    public static float r = 225f;  
    public static bool play = true;  
    void OnTriggerEnter (Collider x){  
        if(x.tag == "FireWall"){  
            Debug.Log ("fire");  
            Born();  
        }  
    }  
  
    // Update is called once per frame  
    void Update () {  
        if(play)  
        {  
            if(!Light.light)  
            {  
                if(((transform.position.x  
sphere.position.x)*(transform.position.x - sphere.position.x) +  
(transform.position.z - sphere.position.z)*(transform.position.z -  
sphere.position.z)) > r)  
                {  
                    Debug.Log ("dead");  
                    Born ();  
                }  
            }  
        }  
        else
```

```
        {
            Debug.Log ("dead");
            Born ();
        }
    }
    else if(Light.light)
    {
        if((((transform.position.x
sphere.position.x)*(transform.position.x - sphere.position.x) +
(transform.position.z - sphere.position.z)*(transform.position.z -
sphere.position.z)) < 25)
        {
            Debug.Log ("dead");
            Born ();
        }
    }
}

void Born()
{
    player.transform.position=new
Vector3((float)Reborn.position_x, (float)Reborn.position_y, (float)Rebo
rn.position_z);
    sphere.transform.position=new
Vector3((float)Reborn.position_x, (float) (Reborn.position_y+20f), (floa
t)Reborn.position_z);
}
}
```

9.11 机关：普通开关（GravityTrigger.cs）

```
using UnityEngine;
using System.Collections;

public class GravityTrigger : MonoBehaviour {

    public Transform player;
    public Transform gate;
    public float speed = 0.001f;
    void Start () {

    }

}
```

```
// Update is called once per frame
void OnTriggerStay(Collider c)
{

    if(c.transform.gameObject==player.transform.gameObject&&gate.transfo
rm.position.y>=-10f)
    {
        gate.transform.Translate
(Vector3.down*Time.deltaTime*0.01f*speed, Space.World);
    }
}
/*
void OnTriggerExit(Collider c)
{
    if(c.transform.gameObject==player.transform.gameObject)
    {
        contact = false;
        gate.transform.Translate
(Vector3.down*Time.deltaTime*0f, Space.World);
        //print ("aaaa");
        //gate.transform.Translate
(Vector3.down*Time.deltaTime*0, Space.World);
    }
}
}*/
}
```

9.12 机关：第一关最终开关（FinalTrigger.cs）

```
using UnityEngine;
using System.Collections;

public class FinalTrigger : MonoBehaviour {

    public Transform player;
    public Transform lighttrigger;
    public Transform gate1;
    public Transform gate2;
    public Transform sphere;
    public Transform light1;
    public Transform light2;
    public Transform light3;
    public Transform light4;
```

```
public float speed = 0.001f;
public float dis= 25f;//作用距离
float sphere_distance;

// Update is called once per frame
void OnTriggerStay(Collider c)
{
    sphere_distance=Vector3.Distance(lighttrigger.transform.position, sphere.transform.position);
    if (c.transform.gameObject==player.transform.gameObject)
    {
        light1.transform.gameObject.SetActive(true);
        light2.transform.gameObject.SetActive(true);
    }
    if (sphere_distance<=dis)
    {
        light3.transform.gameObject.SetActive(true);
        light4.transform.gameObject.SetActive(true);
    }
    else
    {
        light3.transform.gameObject.SetActive(false);
        light4.transform.gameObject.SetActive(false);
    }

    if(c.transform.gameObject==player.transform.gameObject&&sphere_distance<=dis)
    {

        if(gate1.transform.position.y>=-10f&&gate2.transform.position.y>=-10f)
        {
            gate1.transform.Translate
            (Vector3.down*Time.deltaTime*2000f*speed, Space.World);
            gate2.transform.Translate
            (Vector3.down*Time.deltaTime*2000f*speed, Space.World);
        }
    }
}
void OnTriggerExit(Collider c)
{
    if (c.transform.gameObject==player.transform.gameObject)
```



```
{
    light1.transform.gameObject.SetActive(false);
    light2.transform.gameObject.SetActive(false);
}
}
```

9.13 机关：光控开关（Lightcontrol.cs）

```
using UnityEngine;
using System.Collections;

public class Lightcontrol : MonoBehaviour {
    public Transform player;//定义一个人物的 Transform
    public Transform gate;
    public float speed=0.001f;
    public float dis= 25f;//作用距离
    float player_distance;//player 距离
    // Update is called once per frame
    void Update () {

        player_distance=Vector3.Distance(this.transform.position,player.trans
sform.position);
        //Debug.Log (player_distance);
        //触发门
        if (player_distance<=dis)
        {
            if(gate.transform.position.y>=-10f)
            {
                gate.transform.Translate
(Vector3.down*Time.deltaTime*2000f*speed,Space.World);
            }

        }
    }
}
```

9.14 light.cs

```
using UnityEngine;
using System.Collections;
```

```
public class Light : MonoBehaviour {
    public static bool light = false;
    // Update is called once per frame
    void Update () {
        if(Input.GetKeyDown (KeyCode.F))
        {
            light = !light;
            ChangeLight (light);
        }
    }
    void ChangeLight(bool light)
    {
        if(light)
        {
            Death.r = -1f;
            gameObject.light.spotAngle = 30f;
            gameObject.light.range = 50f;
        }
        else
        {
            Death.r = 225f;
            gameObject.light.spotAngle = 100f;
            gameObject.light.range = 30f;
        }
    }
}
```

9.15 createball.cs

```
using UnityEngine;
using System.Collections;

public class creatball : MonoBehaviour {
    public static bool play = false;
    public GameObject prefab;
    public Transform c;
    public Transform target;
    GameObject g;

    void OnClick() {
        play = !play;
        if(play)
        {
```

```
        MoveControl.hight = 0.07f;
        c.position = c.position;
        g = Instantiate (prefab, c.position+new
Vector3(0f, 20f, 0f), target.rotation) as GameObject;
        switch(ColorChange.color){
        case 0:g.light.color = Color.green;break;
        case 1:g.light.color = Color.red;break;
        case 2:g.light.color = Color.yellow;break;
        }
        MoveControl.play = false;
        Death.play = false;

    }
    else
    {
        MoveControl.play = true;
        Death.play = true;
        MoveControl.hight = 0.05f;
        Destroy (g.gameObject);
    }
}

public void ReSet(){
    MoveControl.play = true;
    Death.play = true;
    MoveControl.hight = 0.05f;
    Destroy (g.gameObject);
}
}
```

10. 开发路径规划

10.1 UI 素材和界面设计图制作

UI 设计和界面设计图以及标题画面完全原创。小组内成员自己用 Photoshop 设计和实现。

10.2 原画绘制、扫描、上色

原画均为小组成员手绘，用“扫描全能王”软件扫描为电子版。用 sketches 上色。最终用 PS 修改和定稿。

10.3 模型修改、动画制作

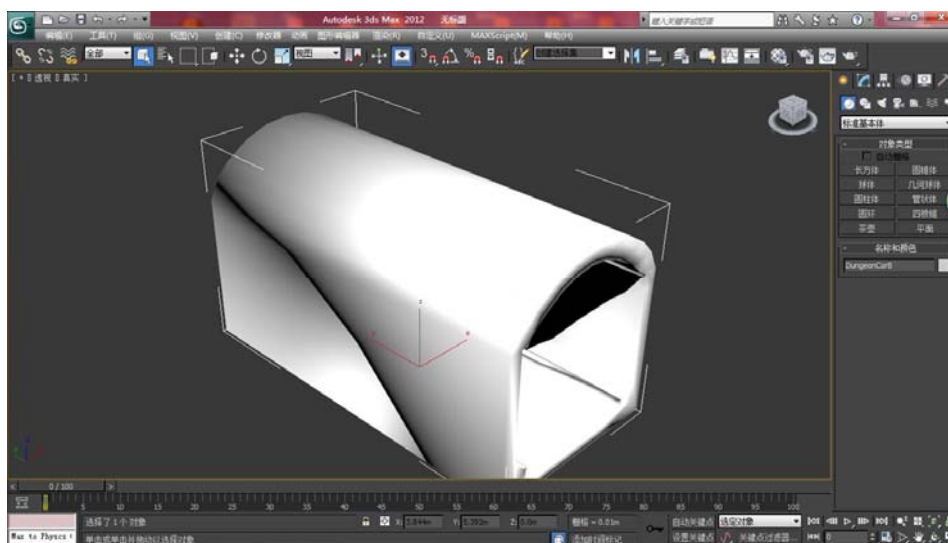


图 10.1

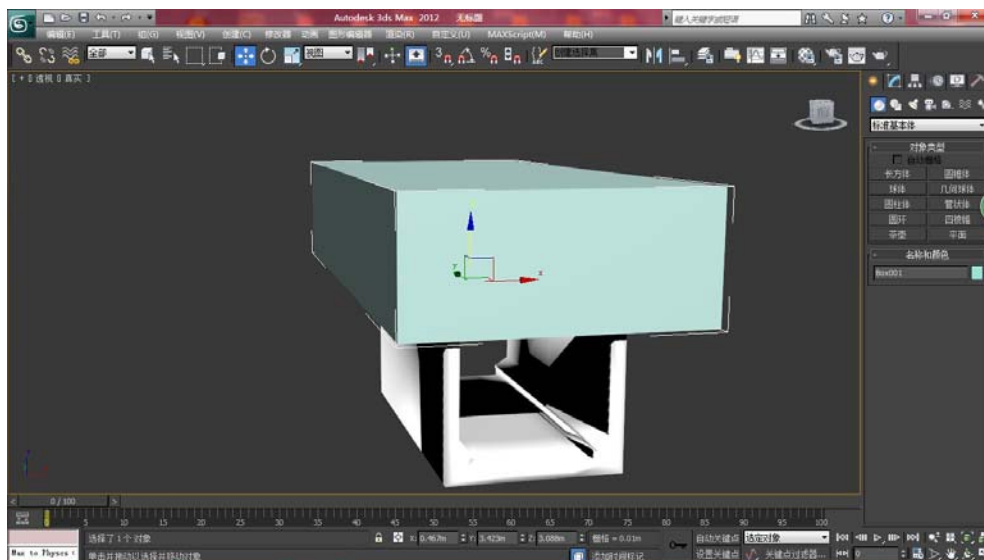


图 10.1

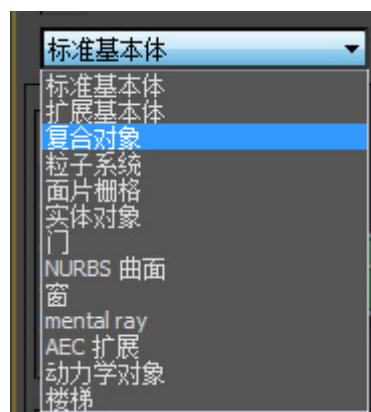


图 10.3

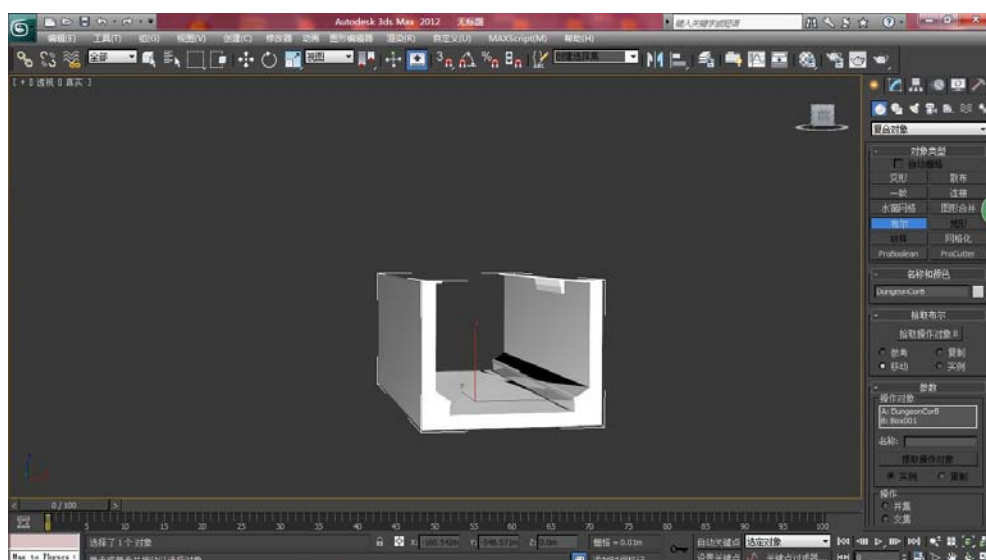


图 10.4 修改场景模型

模型素材从网上搜集得到。小女孩的模型经过 3DS MAX2012 修改得到，并绑定骨骼，制作走路的动画。第二关的场景模型经过修改。

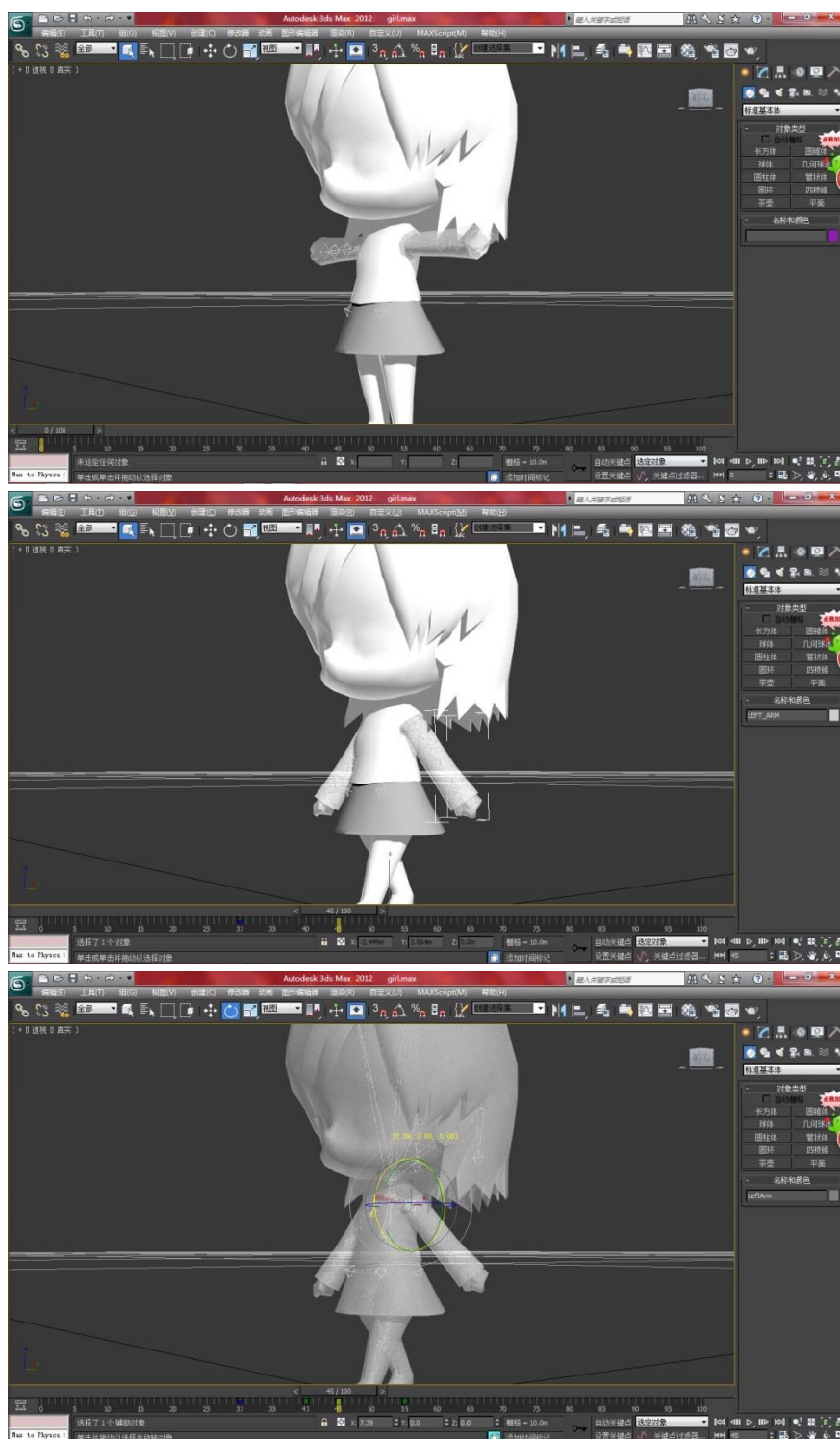


图 10.5 修改小女孩的模型和添加骨骼

10.4 音效处理

音效均出自“魔王魂”BGM 素材网站。部分素材经过 Au 软件特效处理。

10.5 游戏制作和分工

策划：乌若尧

程序：张思语

原画：胥玲

扫描、上色：乌若尧

剧本：陈旭韩

界面设计：乌若尧

场景搭建：张思语

3D 建模和动画：乌若尧、张思语

11. 游戏系统测试

以下为主要游戏界面的测试截屏：



图 11.1 标题界面测试截屏

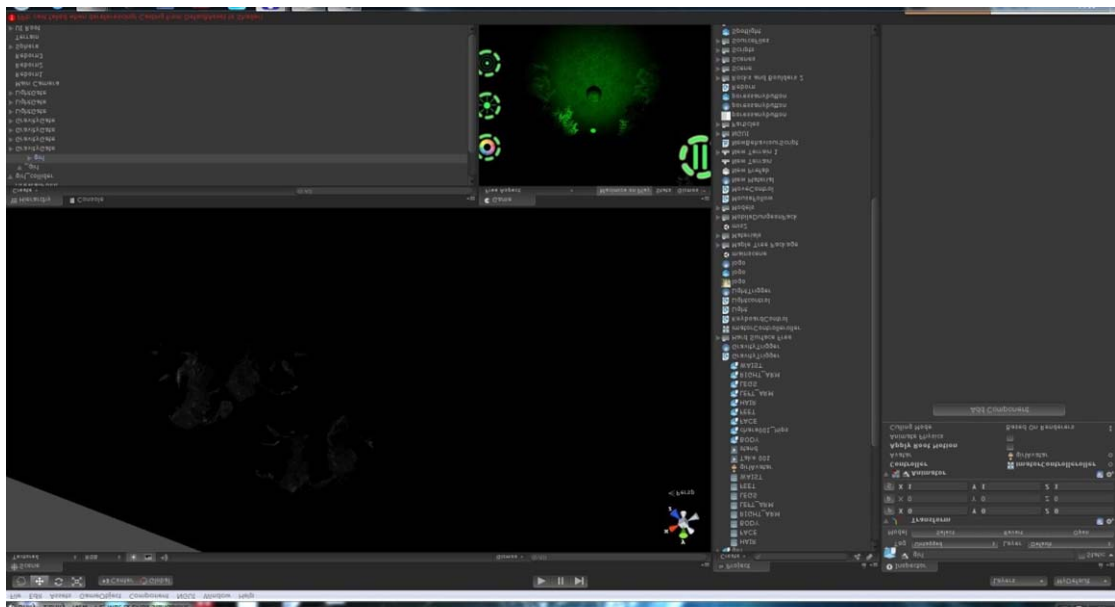


图 11.2 第一关测试

:



图 11.3 角色移动测试

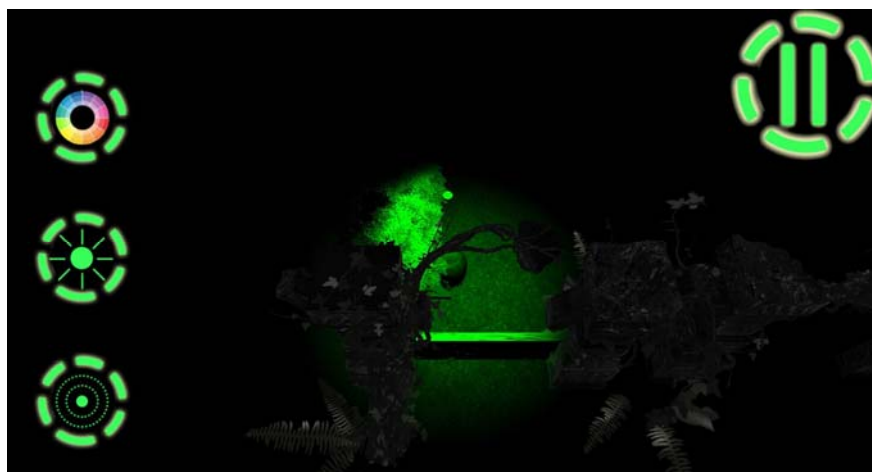


图 11.4 机关测试

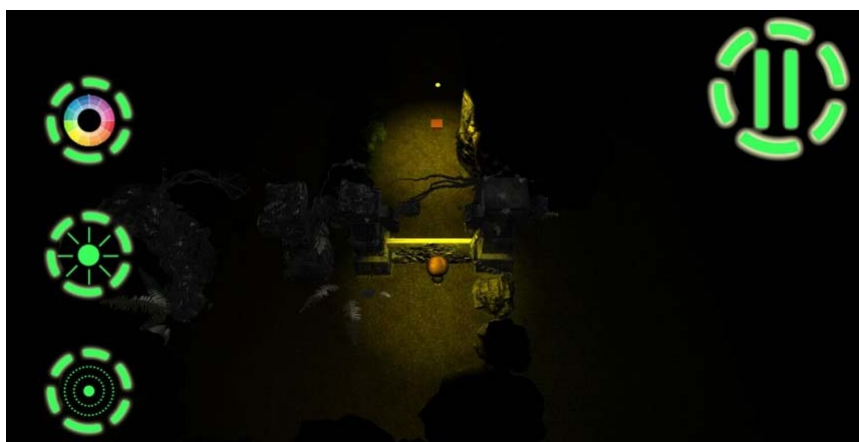


图 11.5 灯光测试



图 11.6 灯光测试

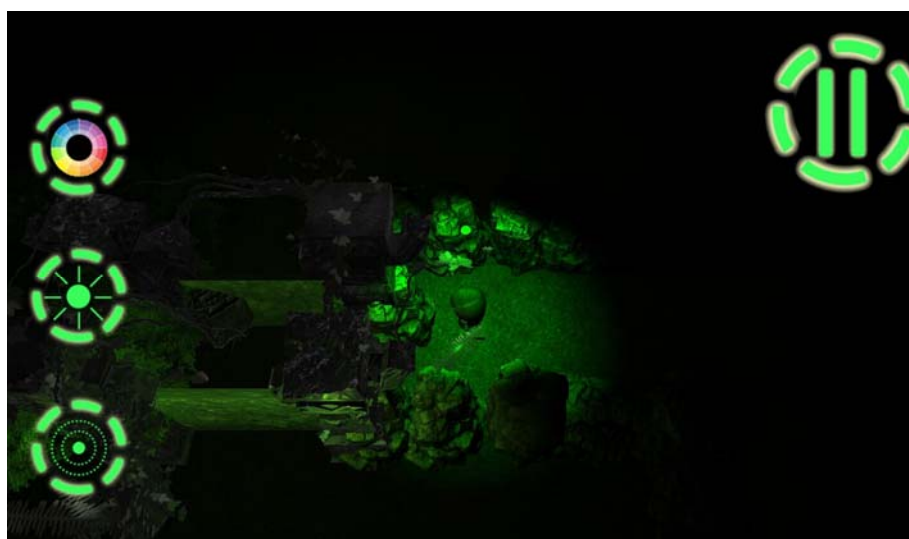


图 11.7 灯光测试



图 11.8 第一关最后机关测试



图 11. 死亡界面