**鲲之源概要设计**

**修订历史记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <6/11/2018> | <1.0> | <概要设计 > | <林智威> |
| <6/13/2018> | <1.1> | <添加了一些原型> | <林智威> |
| <6/13/2018> | <1.2> | <修改一些功能需求> | <王堰楠> |
| <6/13/2018> | <1.3> | <修改了功能需求> | <林智威> |
| <6/13/2018> | <1.4> | <修改了功能需求> | <林智威> |
| <6/13/2018> | <1.5> | <修改了功能需求> | <林智威> |
| <6/15/2018> | <1.6> | <更正一些功能需求> | <丘小霞> |
| <7/12/2018> | <1.7> | <更正一些功能需求> | <庞绮琛> |

本概要设计目的在于明确说明《鲲之源》游戏各功能的实现方式，指导团队进行编码，并解决实现该系统的程序模块设计问题。包括如何把该系统划分成若干个功能模块、决定各个功能模块间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。

在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明。

本文档的预期读者：游戏开发人员（包括美术）。

**1 概述**

**1.1系统简述**

该软件是一款运行在PC端的角色扮演类（RPG）游戏。

**1.2软件设计目标**

主要的实现目标是完成策划中的三个场景，能实现基本的操作，能完成游戏的整体流程，突显游戏的剧情。

主要实现的功能：

1. 游戏的序章：这一模块主要通过动画来实现，引导玩家进入游戏。用户的主要操作有，1. 在游戏开始界面UI上选择继续、开始游戏、设置、查看制作团队，2. 选项界面调节背景音量、音效。

以下的模块会有大多数重复实现功能：

1. 游戏的第一章：第一章是用户真正开始操作的模块，主要实现的功能是：1. 玩家的基本操作，跑，跳，走，使用道具，2. 浮石在一定时间过后掉落，3. 任务失败时，用户回到检查点。4. 游戏的视角，5. 场景物体
2. 游戏的第二章：1. 人物躲避障碍
3. 游戏的第三章：1. 玩家控制鹰的飞行，

**1.3非功能需求**

1. 游戏体验：这是本项目最重要的非功能需求，我们在实现了各种功能后，从用户的角度上讲，这种功能的操作性，即可用性、易用性，需要得到最大的保证。

2. 性能要求：游戏启动过程、运行过程的流畅度。

3. 可靠性：游戏不会中途崩溃，或者在崩溃后能恢复正常。

**1.4参考资料**

《鲲之源游戏策划》

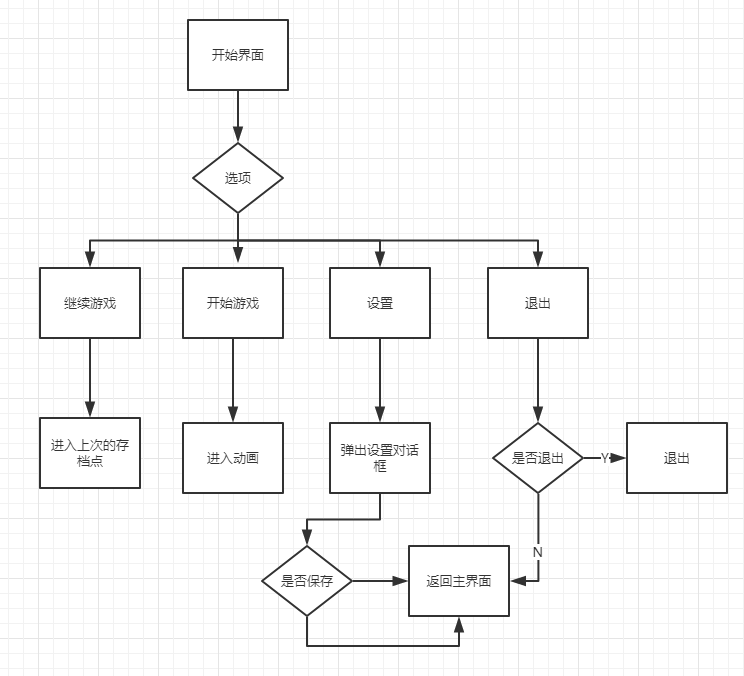
**2 术语表**

|  |  |  |
| --- | --- | --- |
| 开发工具 | Unity3D |  |
| 开发语言 | C# |  |
|  |  |  |

**• 3 设计概述**

**3.1 场景**

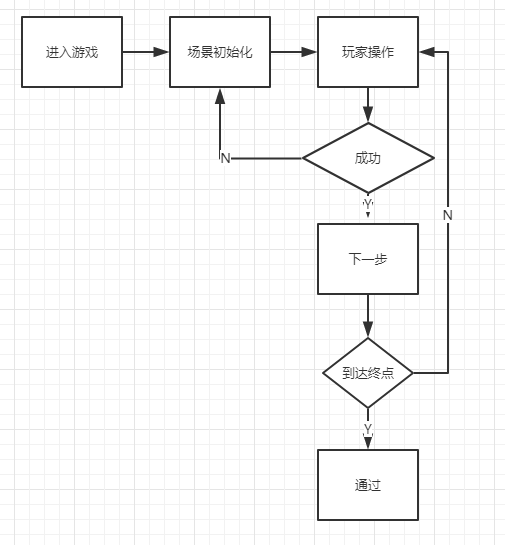
开始界面



序章场景

（无）

第一章场景



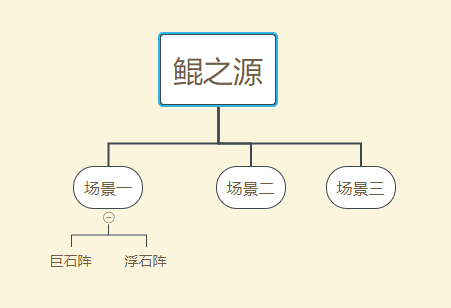
第二章场景

同上

第三章场景

同上

**3.2 系统结构设计**



这部分要求提供高层系统结构（顶层系统结构、各子系统结构）的描述，使用方框图来显示主要的组件及组件间的交互。最好是把逻辑结构同物理结构分离，对前者进行描述。别忘了说明图中 用到的俗语和符号。

1.系统逻辑架构（软件的技术架构）

2.系统物理架构 （物理设备架构图：软件如果部署到物理设备上）

3.系统数据模型（系统逻辑数据模型）

**3.3 功能需求**

游戏包括的范围：开始、序章、第一章、第二章、第三章，游戏范围按场景划分，部分场景的功能会有类似。

|  |  |
| --- | --- |
| 模块 | 需求 |
| 开始界面 | 开始游戏，播放序章动画 |
| 继续游戏，进入上次的存档点 |
| 退出，是否确认退出，确定或取消 |
| 制作团队，查看本游戏的制作团队 |
| 设置 | 音量设置，主音量控制所有音量，背景音乐控制背景音乐音量，音效音量 |
|  |
| 存档 | 用户的存档点都在当前关卡的复活点，不设手动存档的功能。 |
| 视角 | 视角能随着用户鼠标的移动在一定范围内改变。 |
| 序章 | 播放动画，动画结束进入第一章场景一的开头地点 |
| 第一、第二、第三共同的 | 按键，弹出菜单，可以选择继续游戏、退出关卡、设置。 |
| 按键，拾取道具 |
| 玩家回到主界面时，自动存档，存档点为当前关卡的复活点 |
| 体力值耗尽时，不能操控（包括走动），需要静止不动恢复体力 |
| 失败时，可以返回检查点 |
| 第一章 | 走动，不消耗体力 |
| 跑动，消耗体力值 |
| 跳上石柱，消耗体力值 |
| 跳跃障碍，消耗体力值 |
| 使用道具，不消耗体力值 |
| 蓄力跳上浮石，消耗体力值，根据蓄力时长调整消耗的体力 |
| 第二章 | 播放过场动画，动画结束时进入第二章的开头地点 |
| 左右控制人物，躲避障碍，消耗体力值 |
| 起跳躲避障碍，消耗体力值 |
| 第三章 | 播放过场动画，动画结束时第二关的开头地点 |
| 走动，不消耗体力值 |
| 跑动，消耗体力值 |
| 攀爬，消耗体力值 |
| 跳跃躲避障碍，消耗体力值 |
| 使用工具，不消耗体力值 |

**4系统接口设计**

**4.1用户接口**

用户接口主要有：开始界面，确定开始界面，继续游戏界面，选项设置界面，确认退出界面，放弃关卡界面，游戏界面

开始界面：



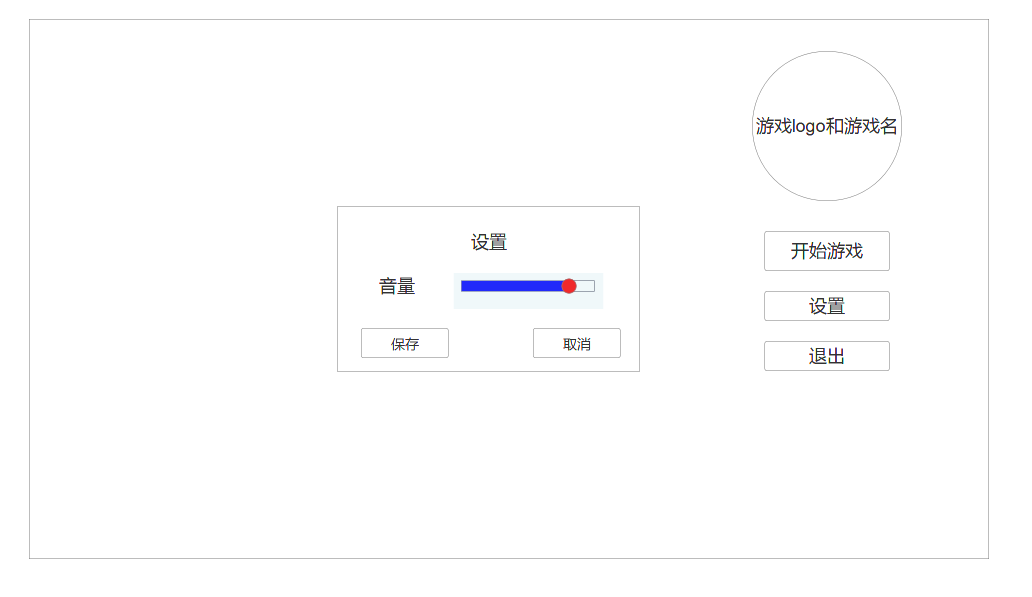
确定开始界面：（背景虚化）



继续游戏界面：



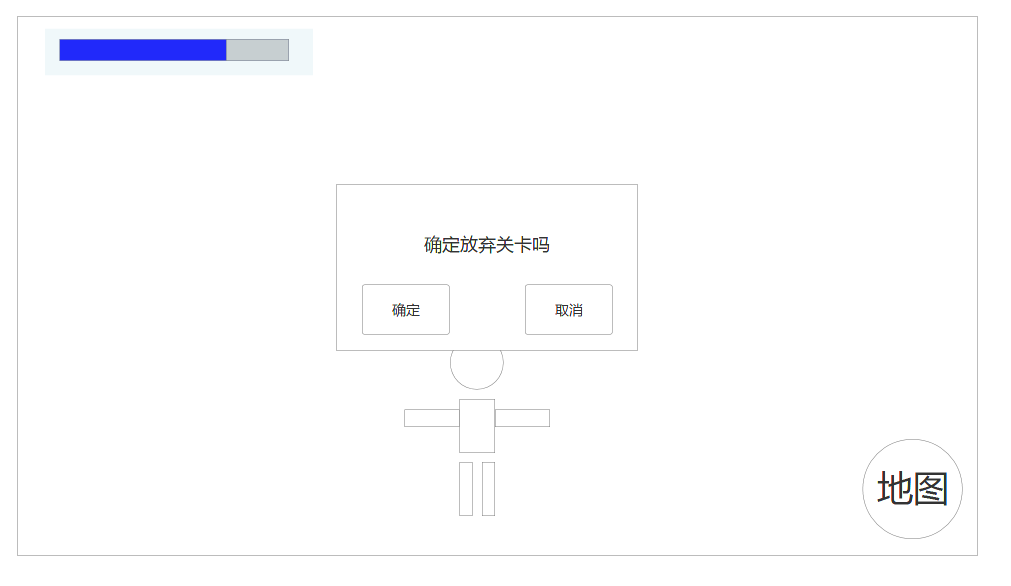
选项设置界面：（背景虚化）



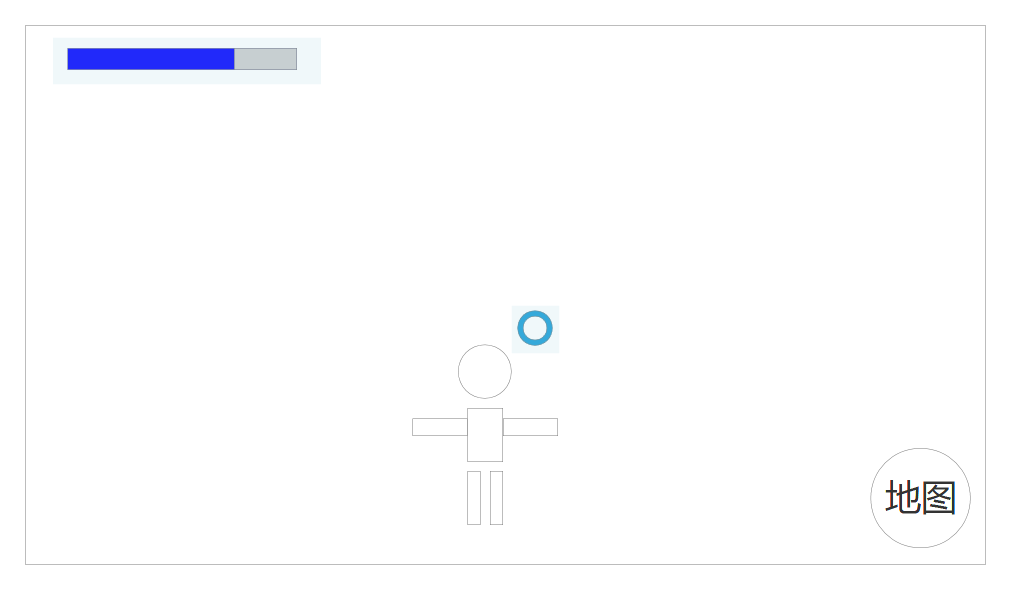
确认退出界面：（背景虚化）



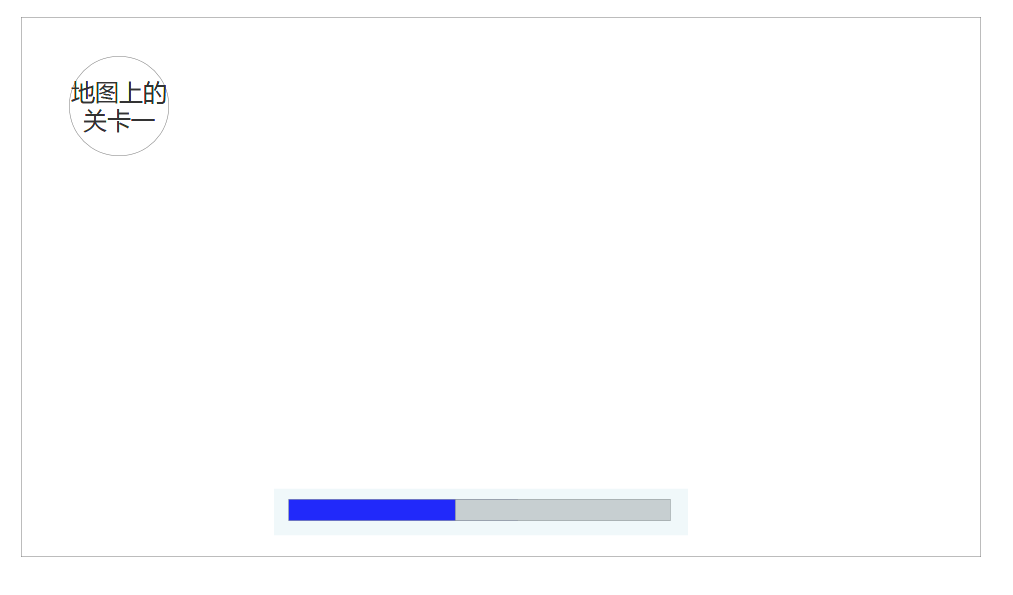
放弃关卡界面：（背景虚化）



游戏界面：



地图加载：



地图：



**4.2 内部接口**

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传递。接口传递的信息将是以数据结构封装了的数据，以参数形式或返回值形式在各模块之间传输。

**4.3 外部接口**

单机游戏，没有数据库、服务器等外部接口。

**5 约束和假定**

描述系统设计中最主要的约束，这些是由客户强制要求并在需求说明书写明的。说明系 统是如何来适应这些约束的。 实现的语言和平台也会对系统有约束，同

样在此予以说明。 对于因选择具体的设计实现而导致对系统的约束，简要地描述你的想法思路，经过怎么 样的权衡，为什么要采取这样的设计等等。

|  |  |  |
| --- | --- | --- |
|  | 详细 | 备注 |
| 约束 | 1. 开发使用的游戏引擎是Unity3D。 2. 开发使用的编程语言是C#。 3. 本项目将会在6周时间内完成，时间跨度为6.4~7.13。 4. 工作时间为9:00AM~6:00PM，每周星期一到星期六。 | Unity3D是现在主流的游戏开发引擎，其使用C#编辑脚本，实现各种需要的功能。Unity3D学习成本较低，能在短时间内开发出成熟的游戏。 |
| 假定 | 该游戏适应 Windows 10系统，并向下兼容Windows 7 |  |

**6非功能性设计**

**6.1 性能**

在95％的情况下，一般时段响应时间不超过1.5秒。

在推荐配置环境下：开始游戏响应时间在2秒内，打开设置等菜单响应时间1秒内，加载游戏场景响应时间3秒内。

**6.2 安全**

严格权限访问控制，玩家只能在设置菜单中修改游戏声音等属性，人物体力值、障碍物等设置玩家不能修改。

**6.3 可靠性**

对输入有提示，数据有检查，防止数据异常。

能处理系统运行过程中出现的各种异常情况，如：玩家操作错误、输入非法数据、资源加载失败等，能正确的处理，恰当的回避。

**6.4 兼容性**

游戏支持windows7及以上操作系统。

**6.5 易用性**

90%的用户在1小时内能够了解并掌握本游戏的基本操作和玩法。

**6.6 可用性**

在95%的故障中，游戏最多需要10秒重启。

游戏中出现错误重启游戏后，能记录到最近一次的存档点继续游戏。

**6.7 可测试性**

交付的游戏必须通过单元测试。

**6.8 可维护性**

90%的BUG修改时间不超过1个工作日，其他不超过2个工作日。