

本科毕业论文（设计）

概要设计说明书

|  |  |
| --- | --- |
| **学 生 姓 名** | **杨明鑫** |
| **学号** | **2015081063** |
| **专业** | **软件工程** |
| **年级班级** | **2015级 2班** |
| **指导教师** | **唐聃（教授）** |
| **所在学院** | **软件工程学院** |
| **提交日期** | **2019年4月19日** |

2019 年 4 月

成都信息工程大学 软件工程学院

目录

[1 引言 1](#_Toc10157792)

[1.1 编写目的 1](#_Toc10157793)

[1.2 背景 1](#_Toc10157794)

[1.3 术语 1](#_Toc10157795)

[1.4 参考资料 1](#_Toc10157796)

[2 总体设计 1](#_Toc10157797)

[2.1 系统体系结构 1](#_Toc10157798)

[2.2 系统功能结构 2](#_Toc10157799)

[2.3 运行环境 2](#_Toc10157800)

[2.3.1 硬件环境 2](#_Toc10157801)

[2.3.2 软件环境 2](#_Toc10157802)

[2.4 系统的关键技术 2](#_Toc10157803)

[3 功能模块设计说明 3](#_Toc10157804)

[3.1 功能模块列表 3](#_Toc10157805)

[3.2 地图管理 3](#_Toc10157806)

[3.2.1 模块编号和功能描述 3](#_Toc10157807)

[3.2.2 操作者 3](#_Toc10157808)

[3.2.3 与本模块相关的码表和表 3](#_Toc10157809)

[3.2.4 界面设计与说明 4](#_Toc10157810)

[3.2.5 输入信息 4](#_Toc10157811)

[3.2.6 输出信息 4](#_Toc10157812)

[3.2.7 算法 4](#_Toc10157813)

[3.2.8 处理流程 5](#_Toc10157814)

[3.2.9 类设计 5](#_Toc10157815)

[3.3 道具管理 6](#_Toc10157816)

[3.3.1 模块编号和功能描述 6](#_Toc10157817)

[3.3.2 操作者 6](#_Toc10157818)

[3.3.3 与本模块相关的码表和表 6](#_Toc10157819)

[3.3.4 界面设计与说明 6](#_Toc10157820)

[3.3.5 输入信息 7](#_Toc10157821)

[3.3.6 输出信息 7](#_Toc10157822)

[3.3.7 算法 7](#_Toc10157823)

[3.3.8 处理流程 7](#_Toc10157824)

[3.3.9 类设计 8](#_Toc10157825)

[3.4 角色控制 9](#_Toc10157826)

[3.4.1 模块编号和功能描述 9](#_Toc10157827)

[3.4.2 操作者 9](#_Toc10157828)

[3.4.3 与本模块相关的码表和表 9](#_Toc10157829)

[3.4.4 界面设计与说明 9](#_Toc10157830)

[3.4.5 输入信息 9](#_Toc10157831)

[3.4.6 输出信息 9](#_Toc10157832)

[3.4.7 算法 9](#_Toc10157833)

[3.4.8 处理流程 10](#_Toc10157834)

[3.4.9 类设计 10](#_Toc10157835)

[3.5 指令管理 11](#_Toc10157836)

[3.5.1 模块编号和功能描述 11](#_Toc10157837)

[3.5.2 操作者 11](#_Toc10157838)

[3.5.3 与本模块相关的码表和表 12](#_Toc10157839)

[3.5.4 界面设计与说明 12](#_Toc10157840)

[3.5.5 输入信息 12](#_Toc10157841)

[3.5.6 输出信息 12](#_Toc10157842)

[3.5.7 算法 12](#_Toc10157843)

[3.5.8 处理流程 13](#_Toc10157844)

[3.5.9 类设计 14](#_Toc10157845)

[3.6 选关界面 15](#_Toc10157846)

[3.6.1 模块编号和功能描述 15](#_Toc10157847)

[3.6.2 操作者 15](#_Toc10157848)

[3.6.3 与本模块相关的码表和表 15](#_Toc10157849)

[3.6.4 界面设计与说明 16](#_Toc10157850)

[3.6.5 输入信息 16](#_Toc10157851)

[3.6.6 输出信息 16](#_Toc10157852)

[3.6.7 算法 16](#_Toc10157853)

[3.6.8 处理流程 17](#_Toc10157854)

[3.6.9 类设计 17](#_Toc10157855)

[3.7 开始界面 18](#_Toc10157856)

[3.7.1 模块编号和功能描述 18](#_Toc10157857)

[3.7.2 操作者 18](#_Toc10157858)

[3.7.3 与本模块相关的码表和表 18](#_Toc10157859)

[3.7.4 界面设计与说明 19](#_Toc10157860)

[3.7.5 输入信息 19](#_Toc10157861)

[3.7.6 输出信息 19](#_Toc10157862)

[3.7.7 算法 19](#_Toc10157863)

[3.7.8 处理流程 19](#_Toc10157864)

[3.7.9 类设计 20](#_Toc10157865)

[3.8 游戏帮助 20](#_Toc10157866)

[3.8.1 模块编号和功能描述 20](#_Toc10157867)

[3.8.2 操作者 20](#_Toc10157868)

[3.8.3 与本模块相关的码表和表 20](#_Toc10157869)

[3.8.4 界面设计与说明 21](#_Toc10157870)

[3.8.5 输入信息 21](#_Toc10157871)

[3.8.6 输出信息 21](#_Toc10157872)

[3.8.7 算法 21](#_Toc10157873)

[3.8.8 处理流程 22](#_Toc10157874)

[3.8.9 类设计 22](#_Toc10157875)

[4 内部接口设计 23](#_Toc10157876)

[4.1 接口1 23](#_Toc10157877)

[4.2 接口2 24](#_Toc10157878)

# 引言

## 编写目的

本概要设计说明书目的在于明确说明“编程机器人”游戏各功能的实现方式,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、决定各个功能模块之间的接口、模块之间传递的信息,以及数据结构、模块结构的设计等。

## 背景

“编程机器人”是一款编程类的益智游戏，主要针对的用户是6~15岁儿童，使用户在娱乐过程中培养逻辑力、思考力和观察力。

## 术语

指令：玩家控制角色做出相应行动的命令。

障碍物：阻拦角色移动的物体。

道具：游戏道具，帮助玩家清除障碍物。

## 参考资料

《Unity Manual 用户手册》 http://www.ceeger.com/Manual/index.html

《Learning hard C#学习笔记》 李志 编

# 总体设计

## 系统体系结构

|  |
| --- |
|  |
| 图2‑1 系统体系结构图 |

## 系统功能结构

|  |
| --- |
|  |

图2‑2 系统功能结构图

## 运行环境

### 硬件环境

1：显示器：要求分辨率在1024×768像素及以上。

2：内存：1 GB RAM

3：硬盘空间：100MB可用空间。

### 软件环境

操作系统：Windows XP及以上。

## 系统的关键技术

本游戏基于Unity3D游戏引擎进行开发，使用C#作为脚本语言，游戏UI采用UGUI界面开发技术。

采用指令模式来设计实现对角色的控制，调用者角色与接收者角色之间没有任何依赖关系，调用者实现功能时只需调用Command 抽象类的execute方法就可以，Command的子类可以非常容易地扩展。

# 功能模块设计说明

## 功能模块列表

表3‑1 功能模块列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块**  **编号** | **模块**  **名称** | **对应需求**  **功能编号** | **所对应**  **需求功能** | **实现**  **优先级** |
| DTGL\_1 | 地图管理 | GAME\_06 | 地图管理 | 高 |
| DJGL\_2 | 道具管理 | GAME\_05 | 道具管理 | 高 |
| JSKZ\_3 | 角色控制 | GAME\_04 | 角色控制 | 高 |
| ZLGL\_4 | 指令管理 | GAME\_03 | 指令管理 | 高 |
| XGJM\_5 | 选关界面 | GAME\_02 | 选关界面 | 高 |
| KSJM\_6 | 开始界面 | GAME\_01 | 开始界面 | 低 |
| YXBZ\_7 | 游戏帮助 | GAME\_07 | 游戏帮助 | 低 |

## 地图管理

### 模块编号和功能描述

DTGL\_1地图管理，主要对场景中道具、障碍物的位置进行初始化或者玩家点击还原按钮时对道具、障碍进行恢复还原。

### 操作者

玩家。

### 与本模块相关的码表和表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **中文注释** | **类型** | | **作用** |
| **码表** | **表** |
| SceneData | 场景数据 | √ |  | 输出 |

### 界面设计与说明

无。

### 输入信息

场景编号：1-5之间整数。

### 输出信息

物体数量：2，大于或等于零的整型。

物体类型：1，1-10之间的整型。

物体位置x轴：0.1，浮点型。

物体位置y轴：0.1，浮点型。

物体位置z轴：0.1，浮点型。

物体旋转x轴：0.1，浮点型。

物体旋转y轴：0.1，浮点型。

物体旋转z轴：0.1，浮点型。

物体缩放x轴：1.0，浮点型。

物体缩放y轴：1.0，浮点型。

物体缩放z轴：1.0，浮点型。

### 算法

脚本激活时，获取各种物体的GameObject(游戏对象)，从json中读取对应场景中应该存在的物体类型及属性，复制对应的GameObject并修改属性。在恢复物体时，检测该物体是否存在，若不存在就复制对应GameObject并修改属性。

### 处理流程

|  |
| --- |
|  |

### 类设计

#### 类图

|  |
| --- |
|  |

#### 类说明

（一）recovery类说明：

1. 功能：地图的初始化和还原。
2. 主要方法：

init() 初始化，获取场景信息

Rec() 判断要还原物体的种类

setSon() 设置物体属性

1. SceneInfo\_Json类说明：
2. 功能：Json数据的读取与修改。
3. 主要方法：

LoadDataFromJson() 从json里面读取数据

SaveDataToJson() 储存数据到json

GetDirectoryState() 获取关卡信息

SetDirectoryState() 存储关卡信息

## 道具管理

### 模块编号和功能描述

DJGL\_2道具管理，主要对管理道具的使用或者获得。

### 操作者

玩家。

### 与本模块相关的码表和表

无。

### 界面设计与说明

无。

### 输入信息

道具类型编号：1-5整型。

### 输出信息

无。

### 算法

获得道具时根据道具种类，对相应道具的数量+1。

使用道具时根据道具种类，调用相应的方法(如UseRedKey())，首先判断是否有道具，如果有，从角色的位置向正前方发射一条射线，检查前方是否有对应的障碍物，如果有就清除障碍物。

### 处理流程



### 类设计

#### 类图

#### 类说明

1. tools类说明：
2. 功能：控制道具的获取和使用。
3. 主要方法：

getTools() 获取道具

useTools() 使用道具

UseRedKey() 使用道具的具体实现

UseYellowKey()

UseAxe()

UseHammer()

UseBoard()

## 角色控制

### 模块编号和功能描述

JSKZ\_3角色控制，控制移动角色。

### 操作者

玩家。

### 与本模块相关的码表和表

无。

### 界面设计与说明

无。

### 输入信息

玩家发出的指令。

### 输出信息

角色与指令对应的动作。

### 算法

通过调用Invoker类的AddNextOp方法来添加命令，命令使用List来保存，通过调用Executecmd方法来调用对应命令的Execute方法控制角色做相应动作。

### 处理流程

### 类设计

#### 类图

#### 类说明

1. Invoker类说明：
2. 功能：执行和添加命令。
3. 主要方法：

Restore() 清空命令，还原角色状态

Executecmd() 执行命令

AddNextOp() 添加命令

1. Recerver类说明：
2. 功能：角色具体动作的实现。
3. 主要方法：

MoveUp()

MoveDown()

MoveLeft()

MoveRight()

UseRedKey()

UseYellowKey()

UseAxe()

UseHammer()

UseBoard()

## 指令管理

### 模块编号和功能描述

ZLGL\_4指令管理，玩家添加或删除指令、解析命令。

### 操作者

玩家。

### 与本模块相关的码表和表

无。

### 界面设计与说明

玩家通过点击指令图标或者输入指令代码完成操作

### 输入信息

点击指令图片、输入指令代码。

### 输出信息

角色控制命令。

### 算法

点击指令图标后，将对应指令图片克隆并绑定在指令框中作为子物体，解析指令时，依次读取指令框的子物体，根据相应图片信息判断指令。在代码模式中，直接根据输入的代码进行判断指令。

### 处理流程



### 类设计

#### 类图

#### 类说明

1. Running类说明：
2. 功能：解析指令。
3. 主要方法：

Go() 运行指令

CreatePrompt() 创建错误提示条

checkCode() 解析代码

1. AddOrder类说明：
2. 功能：添加图片指令。
3. 主要方法：

init() 初始化

Click() 点击事件

AddObject() 增加指令图标

AddBox() 增加指令输入框

1. DelOrder类说明：
2. 功能：删除图片指令
3. 主要方法：

Click() 点击事件，删除对应指令

AddBox() 添加指令输入框

1. inputTrigger类说明：
2. 功能：输入代码后调整显示框大小
3. 主要方法：

Click() 点击事件，调整面板大小

getRowNum () 获取代码行数

## 选关界面

### 模块编号和功能描述

XGJM\_5选择可进行的关卡。

### 操作者

玩家。

### 与本模块相关的码表和表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **中文注释** | **类型** | | **作用** |
| **码表** | **表** |
| SceneData | 场景数据 | √ |  | 输出 |

### 界面设计与说明

可以选择的关卡为红色球，不可选择关卡为灰色球并有一把小锁。

### 输入信息

场景编号。

### 输出信息

加载对应场景。

### 算法

读取Json文件，获取关卡属性，将不可选择关卡的按钮设为不可以使用，并创建一个子物体（小锁）。点击关卡按钮后加载对应关卡的场景。

### 处理流程

### 类设计

#### 类图

#### 类说明

（一）Choice类说明：

1. 功能：选择关卡。
2. 主要方法：

init()初始化，关卡信息

ck1() 跳转关卡1

ck2() 跳转关卡2

ck3() 跳转关卡3

ck4() 跳转关卡4

ck5() 跳转关卡5

## 开始界面

### 模块编号和功能描述

KSJM\_6选择开始游戏、退出游戏。

### 操作者

玩家。

### 与本模块相关的码表和表

无。

### 界面设计与说明

### 输入信息

无。

### 输出信息

无。

### 算法

点击开始游戏加载选关界面。

### 处理流程



### 类设计

#### 类图



#### 类说明

1. Start\_bt类说明：
2. 功能：选择开始游戏或者退出游戏。
3. 主要方法：

start() 加载选关场景

end() 结束进程

## 游戏帮助

### 模块编号和功能描述

YXBZ\_7显示帮助界面，帮助玩家了解游戏规则。

### 操作者

玩家。

### 与本模块相关的码表和表

无。

### 界面设计与说明

点击帮助按钮，弹出帮助框

### 输入信息

无。

### 输出信息

无。

### 算法

调用GameObject中的SetActive方法来设置帮助框是否可视。

### 处理流程

### 类设计

#### 类图

#### 类说明

1. ClickButton类说明：
2. 功能：控制帮助框。
3. 主要方法：

Help() 显示帮助框

CloseHelp() 关闭帮助框

# 内部接口设计

表4‑1 构件接口列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块名称** | **接口编号** | **接口名称** | **接口类型** | **说明** |
| 地图管理 | 01 | GetDirectoryState | 内部 | 获取关卡信息 |
| 02 | SetDirectoryState | 内部 | 储存关卡信息 |
| … |  |  |  |  |
|  |  |  |  |

## 接口1

1. 接口属性设计

表4‑2 GetDirectoryState接口说明

|  |  |
| --- | --- |
| **接口编号** | 01 |
| **接口名称** | GetDirectoryState |
| **接口说明** | 获取关卡信息 |
| **数据来源** | SceneData.json |
| **调用者** | 地图管理，选关界面 |
| **输入** | 关卡编号 |
| **输出** | 关卡信息 |
| **处理流程** | 至上而下数据匹配 |

1. 接口处理流程图



## 接口2

1. 接口属性设计

表4‑2 SetDirectoryState接口说明

|  |  |
| --- | --- |
| **接口编号** | 01 |
| **接口名称** | SetDirectoryState |
| **接口说明** | 储存关卡信息 |
| **数据来源** | SceneData.json |
| **调用者** | 地图管理，选关界面 |
| **输入** | 关卡信息 |
| **输出** | 无 |
| **处理流程** | 至上而下数据匹配 |

1. 接口处理流程图