“新版数独（New SUDOKU）”项目

软件设计说明书

15211027 李熙

# 1引言

## 1.1编写目的

我之所以要做数独这个小项目，是为了重现经典小游戏，在学习一种新的语言（C#）的同时学习制作图形界面，并提升自己的编程能力。

两个星期，想要系统的学习一种新的语言并且做出个像样的项目，是十分困难的。所以，我并没有系统的学习C#，只是了解了基本的C#语法知识，在实践过程中再学习需要的知识。幸运的是，C#和C++有共通之处，了解了基本知识后，很快就能上手。另外，Visual Studio是一个非常强大的工具，有很多强大的库，为我们的编程提供了许多便利，省去繁琐的步骤，这为项目的完成提供了许多便利。这使得我在第一个星期就完成了这个项目。在这个过程中，自己从构想、设计、实现、debug、优化都是一个人完成，自己对软件有了更深的理解。

## 1.2 设计思路

首先，认识到自己的能力在众多大神中并不是很出彩，那么如何让自己的作品有自己的亮点？我想到，提高作品的“颜值”。所以，这个项目中间的各个控件，都是由我自己制作的图片填充，统一整个界面的风格，给人耳目一新的感觉。

当然，“颜值”只是外在，再高的颜值也要有技术支持。真正到技术实现方面才真正体会到做项目的不易之处：

（1）涉及到许多图片以及题目文档，但是我并没有使用数据库，而是选择使用文件，因为这些数据都是静态的，不用经常改变，所以选择使用方便的文件。对于图片，由于每种数字有多种样式，所以在实现过程中，把他们分类放到各种图片数组中，方便之后的提取查找。

（2）另外，细节决定成败。例如，我的项目中，有一个点击左边题目盘各个picbox的函数，一个简单的点击事件要考虑的事情非常多：点击前图片的状态种类，以及点击之后的图片的状态，不光如此，还要考虑其他图片的变化，一个简单的点击事件就写了将近100行代码。另外，像这样的picBox有81个之多，但是他们都是类似的，应用了事件委托之后才大大减少了代码量。

（3）debug要有耐心。做项目过程中遇到了许许多多之前没有遇到过的bug，感谢助教们的帮助，同时并没有一直依赖助教，发现自己打断点debug最终找到bug也是一件十分有成就的事情。

## 1.3命名规范

1. 类名首字母大写

2. 变量和函数名中首字母小写，其后每个英文单词的第一个字母大写，其他小写。

3. 宏、常量和模板名全部大写。

4. 指针标识符名以p或者Ptr开头

5. 前缀大写

## 1.4术语定义

### 1.4.1 C#编程语言

C#是微软公司发布的一种面向对象的、运行于.NET Framework之上的高级程序设计语言。并定于在微软职业开发者论坛(PDC)上登台亮相。C#是微软公司研究员Anders Hejlsberg的最新成果。C#看起来与Java有着惊人的相似；它包括了诸如单一继承、接口、与Java几乎同样的语法和编译成中间代码再运行的过程。但是C#与Java有着明显的不同，它借鉴了Delphi的一个特点，与COM（组件对象模型）是直接集成的，而且它是微软公司 .NET windows网络框架的主角。

## 1.4.2 C# WinForm开发框架：

Visual C#是微软公司.Ner FrameWork框架中的一个重要组成部分，也是微软公司极力推荐的新一代程序开发语言。WinForm是.Net开发平台中对Windows Form的一种称谓。.Net 为开发WinForm的应用程序提供了丰富的Class Library（类库）。这些WinFrom 类库支持RAD(快速应用程序开发)，这些类库被封装在一个名称空间之中，这个名称空间就是System.Windows.Forms。在此名称空间中定义了许多类，在开发基于.Net的GUI应用程序的时候，就是通过继承和扩展这些类才使得我们的程序有着多样的用户界面。

## 1.5参考资料

《软件工程实用教程》 吕云翔编著 2015年出版

《C#入门经典第五版》 Karli Waston编著 2010年出版

## 1.6相关文档

暂无

## 1.7 版本更新记录

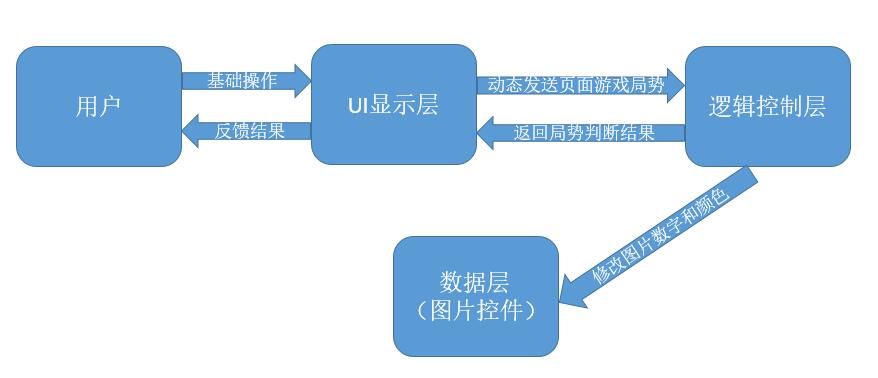
版本更新记录如表1所示。

**表1 版本更新记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 创建者 | 创建日期 | 维护者 | 维护日期 | 维护记录 |
| V1.0 | 李熙 | 2016-7-13 | - | - | - |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 2总体设计

## 2.1总体结构设计



**图1 系统总体结构设计图**

## 2.2 硬件运行环境设计

硬件平台：只支持Window系统

处理器型号：AMD或Intel 1.6GHz以上。

内存要求：1GB以上

## 2.3软件运行环境设计

客户机操作系统：window7及以上版本系统

## 2.4 子系统清单

**表2 子系统清单**

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号 | 子系统名称 | 子系统功能概述 | 子系统关联 |
| SS1 | 数独页面显示子系统 | 包括显示数独游戏的主页面上的数字点数 | 与SS2进行交互：用户点击相应点数后会改变页面显示子系统上的页面显示结果 |
| SS2 | 用户控制子系统 | 用户重做（重新开始）和清除等基础操作 |  |

## 2.5功能模块清单

**表3 功能模块清单**

|  |  |  |  |
| --- | --- | --- | --- |
| 模块编号 | 模块名称 | 模块功能简述 | 模块接口简述 |
| M1-1 | UI系统模块 |  | 各个UI元素对应的功能接口 |
| M1-2 | 用户模块 | 用户的重做，清除，选择游戏难度，查看帮助等基础操作 | 用户使用程序的基础操作接口 |

# 3 模块功能分配

## 3.1 公用模块功能分类

**表4 专用模块功能分配**

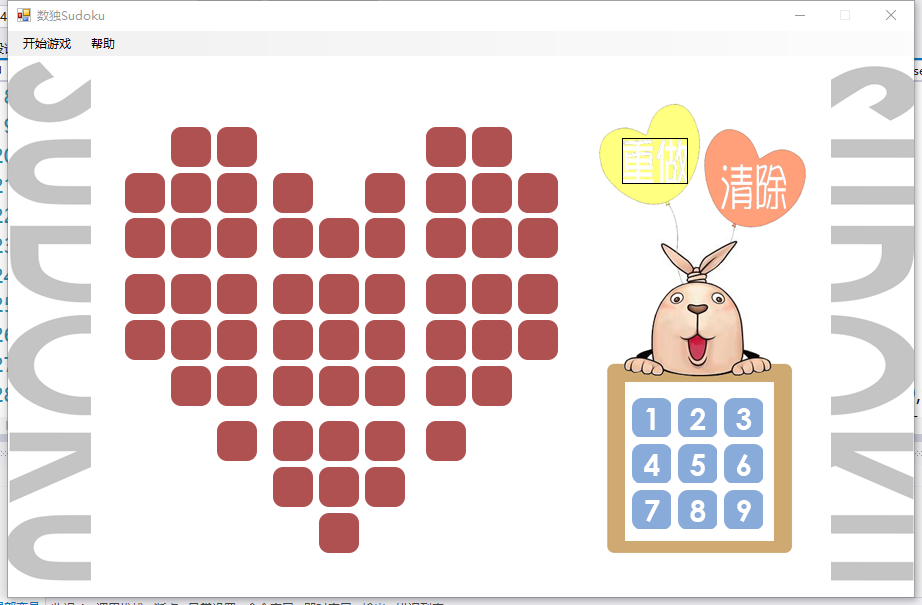
|  |  |  |  |
| --- | --- | --- | --- |
| 模块编号 | 模块名称 | 模块功能简述 | 模块接口简述 |
| G-1 | 页面初始化模块 | 软件初始化相应的控件 | InitPicBox(); 用来初始化所有的PictureBox控件  InitNumber\_c();用来初始化number\_c数组  InitNumber\_o();用来初始化number\_o数组  InitButton();用来初始化所有的Button控件  InitButtonH();用来初始化ButtonH图片数组  InitButtonO();用来初始化ButtonO图片数组    InitEasyQ();初始化简单题目  InitMediumQ();初始化中等题目    InitHardQ();初始化难题 |
| G-2 | 页面响应模块 | 软件绑定用户操作的响应控件 | InitializeComponent();  （里面的控件绑定事件函数暂时未提取出来封装成接口） |

## 3.2 专用模块功能分类

**表5 专用模块功能分配**

|  |  |  |  |
| --- | --- | --- | --- |
| 模块编号 | 模块名称 | 模块功能简述 | 模块接口简述 |
| Z1-1 | 难度选择 | 用户点击下拉菜单中的开始游戏，可以选择不同的游戏难度 | void Easy\_ToolStripMenuItem\_Click(object sender, EventArgs e) 对应简单难度  void Medium\_ToolStripMenuItem\_Click(object sender, EventArgs e) 对应中等难度  void Hard\_ToolStripMenuItem\_Click(object sender, EventArgs e) 对应困难难度 |
| Z1-2 | 查看帮助 | 用户点击查看帮助菜单，弹出查看帮助模块的Form，显示相应信息（实际上在本项目中属于比较小的模块） | Bool SignIn(UserInfo \*Uinfo) |
| Z1-3 | 用户基础操作 | 包括对用户重做和清除操作的捕获和响应 | void button\_reform\_Click(object sender, EventArgs e) 和用户重做操作相对应  void button\_delete\_Click(object sender, EventArgs e)和用户清除操作相对应 |
| Z1-4 | 局势判断模块 | 对游戏当前的局面状况进行实时动态判断的模块 | void tick(object sender, EventArgs e) 设置计时器停止  bool win() 判断游戏当前局势是输或赢  bool isRIght(int x,int y) 判断游戏当前状态是否符合规则，判断用户填入的数字是否正确 |

# 4. 界面设计

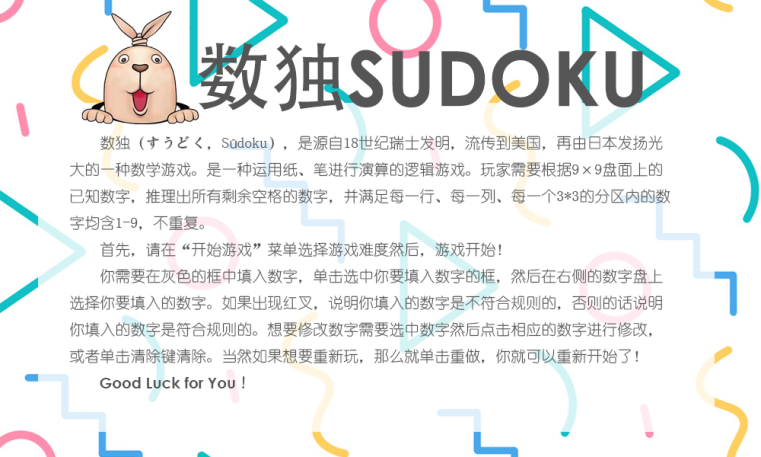


3

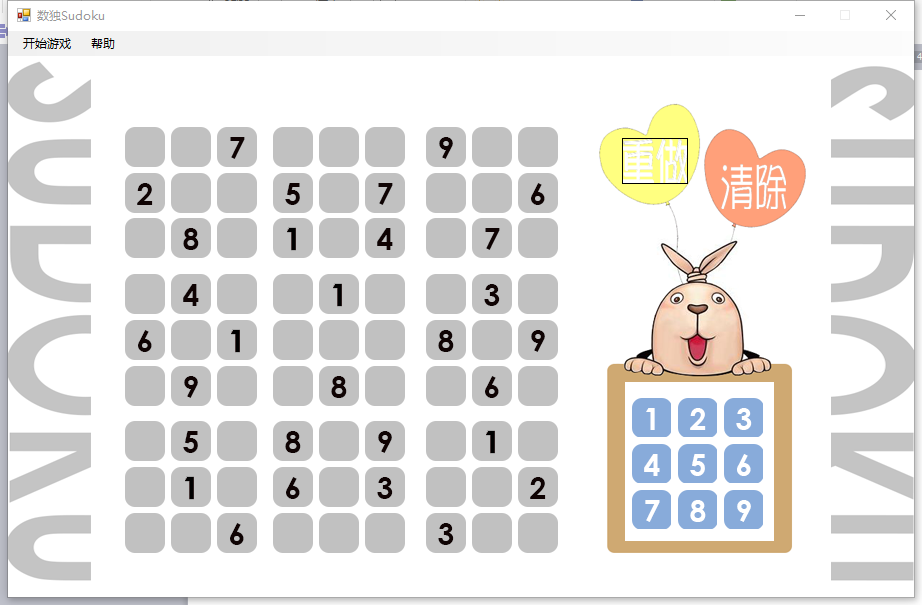
2

1

**图2 游戏开始界面图**



**图3 游戏说明界面图**



6

5

4

**图4 游戏主页面图**

以下分模块按顺序说明各个组件的功能（按图上标注顺序说明）：

## 4.1菜单栏模块

4.1.1“开始游戏”菜单项(模块1)

该菜单项点击后出现如下下拉选项：

游戏难度：易，中，难，测试

用户可以选择不同的游戏难度并开始游戏

4.1.2“帮助(H)”菜单项(模块2)

点击后出现以下下拉项：

(1)使用说明

(2)关于我们：团队成员信息。

## 4.2页面加载模块（模块3）

用户通过点击开始游戏选择了游戏难度以后，应用程序主页面会初始化加载各种控件和素材，并且为控件绑定响应的监听器。

## 4.3 用户操作模块

4.3.1 游戏主界面操作模块（模块4）

用户可以点击主界面中的81个图片块，系统响应用户的点击事件，并做出相应的判断和处理

4.3.2 游戏重做模块（模块5）

用户点击重做按钮游戏会重新开始，系统清除当前界面并重新加载新的游戏界面

4.3.3游戏清除模块(模块6)

用户点击游戏清除按钮，系统立即清除当前选中的框中的数字，重置图片框的状态（颜色等状态）

# 5其他设计

**可靠性**

使用户打开软件后能够正常操作。

**灵活性**

用户可以根据自己的喜好选择不同的游戏难度。

**稳定性**

该软件部署后，在硬件条件和相关支持软件条件没有发生变化的情况下，能够一直保持正常运行，直到软件被升级或取代。

**扩展性**

暂无功能扩展

**故障处理能力**

可能发生的情况为软件崩溃，当软件发生异常情况需要退出或用户在游戏过程中进行非法操作时，通过错误提醒通知用户。

**特殊需求**

暂无。

# 6 系统出错处理设计

## 6.1 出错信息

用户非法操作：

当用户在填入数字的时候填入了非法的数字，那么系统会弹出红叉进行警告，提示用户操作出现错误

## 6.2故障预防与补救

系统提示错误信息，并且提示如何进入正确的操作模式。

# 7 测试计划

采用黑盒测试，对软件的每一子系统进行完全覆盖测试。符合要求后进行整体测试与白盒测试。