摘 要

本次课程设计主要内容是设计一个能够实现抽奖功能的软件。此次课设制作使用了广受游戏开发者们好评的Unity游戏引擎和其所支持的C#语言。制作过程中设计到了动画和音效的播放、图形界面编程、UI设计以及交互设计，在软件开发过程中同时也涉及到文件写入和窗口重绘模式中对UI布局的设置。

**关键词：**抽奖软件，Unity引擎，UI设计，交互设计，C#语言

目 录

[第一章 绪 论 1](#_Toc466640616)

[1.1 Unity游戏引擎介绍和系统配置 1](#_Toc466640617)

[1.2 C#语言简介 1](#_Toc466640618)

[1.3 抽奖软件具体功能介绍 2](#_Toc466640619)

[1.4 抽奖软件具体设计架构 2](#_Toc466640620)

[第二章 抽奖软件开发技术细节 3](#_Toc466640621)

[2.1 数据结构与算法 3](#_Toc466640622)

[2.2](#_Toc466640623)[Unity UI组件 4](#_Toc466640623)

[2.2.1 UI组件介绍 4](#_Toc466640624)

[2.2.2 UI组件在抽奖软件中的应用 7](#_Toc466640625)

[2.3 实现动画和音效 7](#_Toc466640627)

2.4 本章总结 [8](#_Toc466640627)

[第三章 实际效果展示 9](#_Toc466640628)

[第四章 全文总结与改进 1](#_Toc466640635)0

[4.1 全文总结 1](#_Toc466640636)0

[4.2 软件可以改进的地方总结 1](#_Toc466640637)0

[致 谢 1](#_Toc466640638)1

[参考文献 12](#_Toc466640639)

第一章 绪 论

1.1 Unity游戏引擎介绍

Unity 是一款由 Unity Technologies 研发的跨平台2D / 3D 游戏引擎，可用于开发 Windows、MacOS及 Linux 平台的单机游戏，PlayStation、XBox、Wii、3DS 和 任天堂Switch 等游戏主机平台的视频游戏，或是 iOS、Android 等移动设备的游戏。Unity所支持的游戏平台还延伸到了基于 WebGL 技术的 HTML5 网页平台，以及 tvOS、Oculus Rift、ARKit 等新一代多媒体平台。除可以用于研发电子游戏之外，Unity 还是被广泛用于建筑可视化、实时三维动画等类型互动内容的综合型创作工具[1]。

Unity引擎采用的是层级式的综合开发环境，可视化编辑，详细的属性编辑器和动态的游戏预览。Unity创作的作品可以部署到多个平台。采用Direct3D(Windows)，OpenGL(Mac,Windows)和自有的APIs(Wii)图形引擎。Unity内置对Nvidia的PhysX物理引擎支持。音效系统基于OpenAI程序库，可以播放OggVorbis的压缩音效。视频播放采用的是Theora编码。内置Lightmapping和Global illumination。

Unity引擎采用的是GameObject和Component的架构，用简单的组件系统替代了传统开发和编程过程中冗长复杂的继承系统，极大地降低了开发难度，大大提高了开发效率。

Unity拥有全球最大的开发者社区，内部有非常多的详细的资料和文档，有助于开发者们寻求更多更实际的解决方案。

本次使用的是Unity 2018.3.8版本。操作系统为Windows 10，CPU 型号为Inter(R) Core(TM) i7-7700HQ 2.80GHz，显卡NVIDIA GTX1050Ti。

1.2 C#语言简介

C#是微软推出的一种基于.NET框架的、面向对象的高级编程语言。C#以.NET框架类库作为基础，拥有类似Visual Basic的快速开发能力。C#由安德斯·海尔斯伯格主持开发，微软在2000年发布了这种语言，希望借助这种语言来取代Java。C#已经成为Ecma国际和国际标准组织的标准规范[2]。

C#相对于C和C++，在许多方面进行了限制和增强。在C#中只能用于不安全模式中。大多是对象访问通过安全的引用实现，以避免无效的调用。将对象的显式释放禁用，相对地，代替为当不存在被引用时通过垃圾回收器回收。而且C#只允许单一继承。C#相对于C++更加安全，对数组的操作更加方便。C#中用泛型替代模板，而且拓宽了使用范围。但是与此相对地，C#的代码效率比起C++和C要低很多，但是Unity本身对C#语言进行了优化，使得代码效率提高很多。

1.3 抽奖软件具体功能介绍

本次课程设计要求至少能够抽出四种不同奖项，标题与奖项人数由用户定制。抽奖时要有动画与音效，并且抽奖软件布局会随窗口大小改变。最后抽奖结果可以保存。

额外的功能：当鼠标移动到右侧奖池时，鼠标位置会显示奖品名称，奖品剩余数量以及抽取概率。对于奖项人数的修改通过滑动Slider修改。

1.4 抽奖软件具体设计架构

抽奖软件通过创建一个名叫“Observer”的观察者监管整个软件运行情况，调控软件数值和运行逻辑。整个软件由两个脚本文件组成，都挂在观察者之上，一个负责监管游戏，另一个负责实现动画效果，两个脚本间通过几个布尔值进行联络。

在控制脚本中，拥有每个交互式组件的触发逻辑函数，通过Unity自身的UI系统为每个UI组件添加相应的函数来实现界面交互。

控制脚本中还有负责计算的模块，用于计算每个奖项的获奖概率。在抽奖阶段，程序使用Unity自带的随机函数取值。不中奖的概率为50%，中奖概率为50%。具体各个奖项获奖概率由奖品数量动态决定。所以，这个抽奖程序的抽奖机制也很新颖。简单解释就是，当用户没抽到奖时，一切概率不变；当用户抽到奖时，将对应奖项减一，那么与此同时，为了保证获奖概率为50%，所有奖项的被抽到概率都会增加。总体原则是：保证获奖和不获奖的概率都为50%。

最后，所有抽奖记录保存在特定的txt文件中。

第二章 抽奖软件开发技术细节

对于开发一款抽奖软件而言，解决好在使用Unity的过程中遇到的技术细节问题至关重要。本章将把关注点落实在具体的技术细节实现上，细细分析在抽奖软件开发中用到的具体的技术。

2.1 数据结构与算法

此次开发抽奖软件的过程中，考虑到数据的安全性和整体性，将和奖项有关的字段属性封装到了一个结构中，在这个结构中有奖项名称和奖项数量，并且有相关的对外开放的取值和修改接口。在之后的代码中，将奖项结构实例化为四个不同的奖项，从而解决了数据封装问题。

由于观察者要调用不同UI组件的方法，所以Observer必须能够找到相应的组件。考虑到Unity中有两种不同的方法可以实现这个功能，分别是调用Find函数和序列化。GameObject Find(string name)函数从属于类GameObject，从Unity的Hierarchy寻找与string name一样命名的物体，并将其赋给相应的GameObject，这种方法写起来更简单，但是相应地，其代价很高，运行效率较低；而另一种方法使用序列化[SerializeField],如：[SerializeField] private Slider slider1;序列化之后，脚本组件会在视图窗口为该序列化变量提供一个可拖拽赋值的小窗口（如图2-1所示），可以将相应类型的组件拖入该窗口完成赋值，然后代码可以通过使用slider1（上例）作为该赋值的名称，进而调用其内部方法。这种序列化的方法效率更高，所以本次课设基本上都是对所需变量进行序列化，只有个别组件的赋值采用了Find函数。

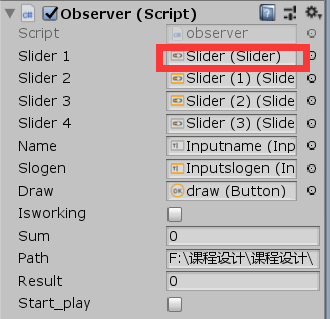


图2-1 为序列化变量拖拽赋值

对于本次抽奖软件的核心部分，也就是抽奖算法而言，我有着一套比较令用户满意的抽奖方案。这个方案的核心是保证每个用户抽奖时抽中的概率是50%。那么与此所匹配的算法逻辑如下：初始时，根据slider组件（可以调节中奖人数的组件）的初始值设置每个奖项中奖的基础概率；根据奖项人数确定随机数范围，这个范围是从1到2\*奖品总个数+1，由于Unity提供的随机数API Random.Range(a,b)的范围是左开右闭集合，所以最大值要+1，之所以最大值是2\*奖品个数是为了保证中奖概率和不中奖概率各一半；接下来处理得到的随机值：当随机值处于1到奖品总数内，将结果置0；如果随机值在奖品总数+1到奖品总数+第一类奖品个数范围内，将结果置1；如果随机值在奖品总数+第一类奖品个数+1到奖品总数+第一类奖品个数+第二类奖品个数范围内，将结果置2；如果随机值在奖品总数+第一类奖品个数+第二类奖品个数+1到奖品总数+第一类奖品个数+第二类奖品个数+第三类奖品个数范围内，将结果置3；最后一种情况置为4；当奖项人数改变时，重新进行上述操作。上述不同的结果对应不同的奖项。

尽管一开始考虑到普通抽奖模式类似于从盒子中摸球：每摸出一个球，不管有没有中奖，都会将这个球移出奖池。这样的抽奖方式虽然很符合现实，但是考虑到计算机内部可以采用不同的抽奖逻辑，于是使用了中奖率比较高的方式，即：保证每个人抽奖的时候中奖概率为50%。现实中抽奖很难保证有这样的结果，除非每次将奖池中不中奖的小球数量进行调整。但对于计算机而言，这种方式还是很容易实现。换一个角度思考这个算法，其实商场轮盘抽奖的情况可能会出现这种情况，就是虽然转到中了某个奖品但是这个奖品已经被抽光了，尽管看起来像（而且之后的抽奖动画也是转盘），但内部逻辑还是不尽相同。

2.2Unity UI 组件

利用Unity开发抽奖软件有一个核心的问题需要解决，那就是人机交互。而对于一般的人机交互方式，通常是通过UI来实现的。

2.2.1 UI组件介绍

2.2.1.1 Canvas

Unity的UI系统的构建类似于画家作画。而Canvas就是画作的画布，一切其他的UI组件必须是某一块Canvas的子物体，必须依附于Canvas上才能正常使用。Canvas是后期摄像机的渲染的对象，而这个抽奖软件的要求之一是在调整窗口大小调整之后UI的布局能够跟着改变，那么就需要设置Canvas的属性。具体调整是：将Canvas Scale组件的UI Scale Mode改为 Scale with Screen Size 然后将Screen Match Mode改为Match Width Or Height，这样，就能动态改变UI布局了。（具体配制如图2-2所示）

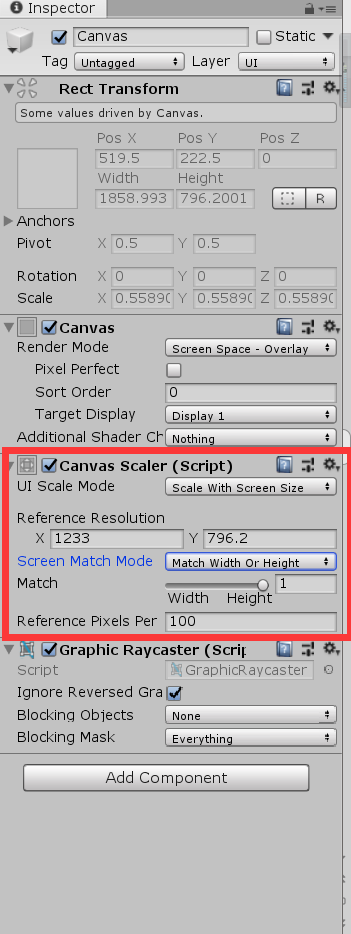


图2-2 Canvas设置

2.2.1.2 Input Field

对于自定义抽奖软件的标题和抽奖人姓名的输入，需要用Input Field组件进行读取。Input Field组件自带消息处理函数用于在输入框中的值改变或者结束输入的处理。Input Field 自带两个子文本，一个文本用于显示输入前的提示语句，另一个文本用于获取用户输入。在本软件中，当用户点击Input Field 时，提示语句文本框内容会被清空。

2.2.1.3 Button

在一般的UI设计中，用途最为广泛的组件就是Button了。Button组件会在用户点击之后触发一个事件。

2.2.1.4 Image

如果想要在Canvas界面上添加一些图形图像，无法直接把诸如bmp或者jpeg的图片拖拽进入Unity当中，要先把原始图片变为Sprite，然后将Image组件作为载体，然后把Sprite赋给Image才能正常显示图片。如果想直接使用原始图片，可以使用Raw Image组件，但是Raw Image组件的判定框不正常，可能影响其他组件的点击判定。

2.2.1.5 Slider

Slider组件是一个可以滑动的条带，上面有一个按钮，可以用鼠标滑动。Slider组件可以根据其上按钮相对于起点的位置和最大值从而换算出相应的值。Slider组件有一个当值改变的时候触发的函数，此时会同步更新奖项结构里面的奖品数，从而实现由与UI交互改变奖项数的功能。（Slider如图2-3所示）

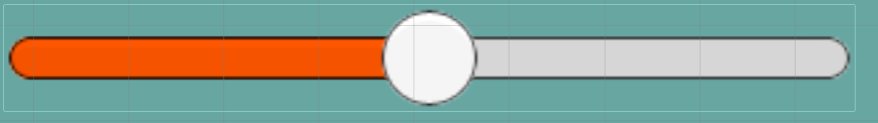


图2-3 Slider

2.2.1.6 组件：Event Trigger

在Unity给出的UI中，事件响应函数是有限的。有时候我们想要在鼠标移动到某个按钮上（Button只有点击触发事件）触发一个事件，这个时候我们如果从底层实现这个功能会比较复杂，而Event Trigger正是为了应对这个情况而被创建的。Event Trigger包含许多不同类型的交互响应函数（如：PointerEnter是处理鼠标移入某UI时触发事件）。（图2-4 Event Trigger）

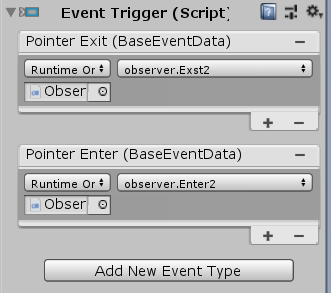


图2-3 Event Trigger

2.2.2 UI组件在抽奖软件中的应用

在实现自定义标题和名字输入时，添加了Input Field；在实现修改人数的功能时，使用了slider；在某些静态的文本，如：显示某奖品的详细信息，使用了Text；在实现某些没有交互的静态图片或者背景图片时，使用了Image；在处理某些只有点击事件的UI时，使用了Button。

在这次课程设计中，添加了额外的功能，就是当鼠标移动到奖池上的特定奖项上时，会在鼠标旁边显示一个文本框，框中显示的是奖项名称、奖品剩余个数和抽到的概率。实现这个功能靠的是一对具有父子关系的文本框实现的。父文本框控制整体大小，子文本框搭配一张纯色图片显示实际内容。使用Event Trigger中的事件函数，当鼠标指针移动到某个奖项时，将准备好的文本框从屏幕外移动到鼠标指针旁；当鼠标指针移开时，将文本框移除屏幕。这样就能实现基于鼠标的信息自动显示。

2.3 实现动画和音效

抽奖软件的动画播放是基于几何变换而形成的。首先创建了一个圆盘，圆盘上分割了八个区域，获奖与不获奖的扇形分隔开来，圆盘中间有一个始终指向正上方的指针。在玩家点击抽奖按钮之后，圆盘开始转动，三秒后圆盘停止，显示最终抽奖结果。抽奖圆盘在动画播放的这三秒之内以一定速度绕圆盘中心旋转，具体调用的方法是：ani.transform.Rotate(Vector3.forward \* 100);Vector3.forward指的是绕z轴旋转，也就是垂直于屏幕的轴。实现动画的关键是控制播放时间。此时要调用携程函数。携程函数是Unity里的一类特殊的函数，通过yield return new WaitForSeconds(seconds)在seconds秒内调用此语句之前的语句，在seconds秒之后调用此语句之后的语句。协程通过StartCoroutine开始调用，通过StopCoroutine结束。所以应当把旋转函数放在yield return之前，每帧调用达到播放动画的功能；将StopCorutine放在yield return之后，实现在seconds秒之后停止播放动画。

一开始想到轮盘应该最后停到对应的奖项上，这样的话旋转时间就不能确定。而且旋转角度也很难确定。所以我想到了一种解决方案：在三秒内使圆盘旋转变得很快，在最后停止的时候，根据抽到奖项的不同，轮盘最后会停在不同的位置。这样就用最简单的方式实现了动画播放功能。

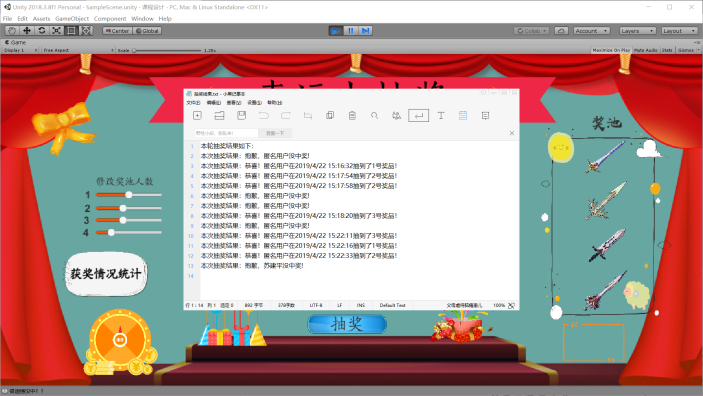
在处理声音播放的时候，需要在Observer上添加Audio Source组件，然后把准备好的音频文件添加到Audio Source上。然后在代码中需要播放声音的地方调用Audio Source的Play函数，并在需要停止的地方调用Stop函数。

2.4 本章小结

本章从数据结构与算法和UI两个方面介绍了开发抽奖软件的详细细节。算法方面简化了抽奖动画播放方式，而且创造了一种新的抽奖方式。UI是一个交互式软件的核心部件，Unity的UI系统比较容易上手，依靠自身的事件处理函数和Event Trigger的时间处理函数可以达到很好的交互体验。

在抽奖动画播放期间，将所有可交互的按键全部禁用，等到动画全部播放完毕，再启用所有按键。

1. 实际效果展示



第四章 全文总结与改善

4.1 全文总结

此次制作的抽奖软件实现了课程设计文档中的所有要求。而且，还额外实现了当鼠标移动到特定奖项上时，展示该奖项的详细信息。在概率分配和计算上面有独特的算法，在动画播放和图形变化上简化了计算机处理，但是同样地达到了效果。抽奖软件的开发是一个需要强交互的设计过程，核心就是UI和用户的交互，Unity的UI系统拥有一套不是特别完备的事件处理函数，但是可以通过添加额外的Event Trigger中的事件处理函数来完善交互功能，从而实现任意的想要的交互效果。最后，所有的抽奖结果会保存在一个txt文件之中。

4.2 软件可以改进的地方总结

美术方面：UI的美术效果可以进一步细化，UI风格可以进一步统一，或者增加一些特效。

动画方面：可以让轮子旋转从慢到快，再从快到慢，细化抽奖动画，从而显得不是特别突兀。

致 谢

本论文的工作是在我的导师邢建川老师悉心指导下完成的，对于我的每一个问题，邢老师都做出了详细的解答，十分感谢。

参考文献

1. <https://zh.m.wikipedia.org/wiki/Unity_(%E6%B8%B8%E6%88%8F%E5%BC%95%E6%93%8E)> 维基百科
2. <https://zh.m.wikipedia.org/wiki/C%E2%99%AF> 维基百科