# 游戏设计实验报告

项目人员：21172710 韩霖

项目名称：台球争霸

项目时间：2020.4

## 项目总体简介：

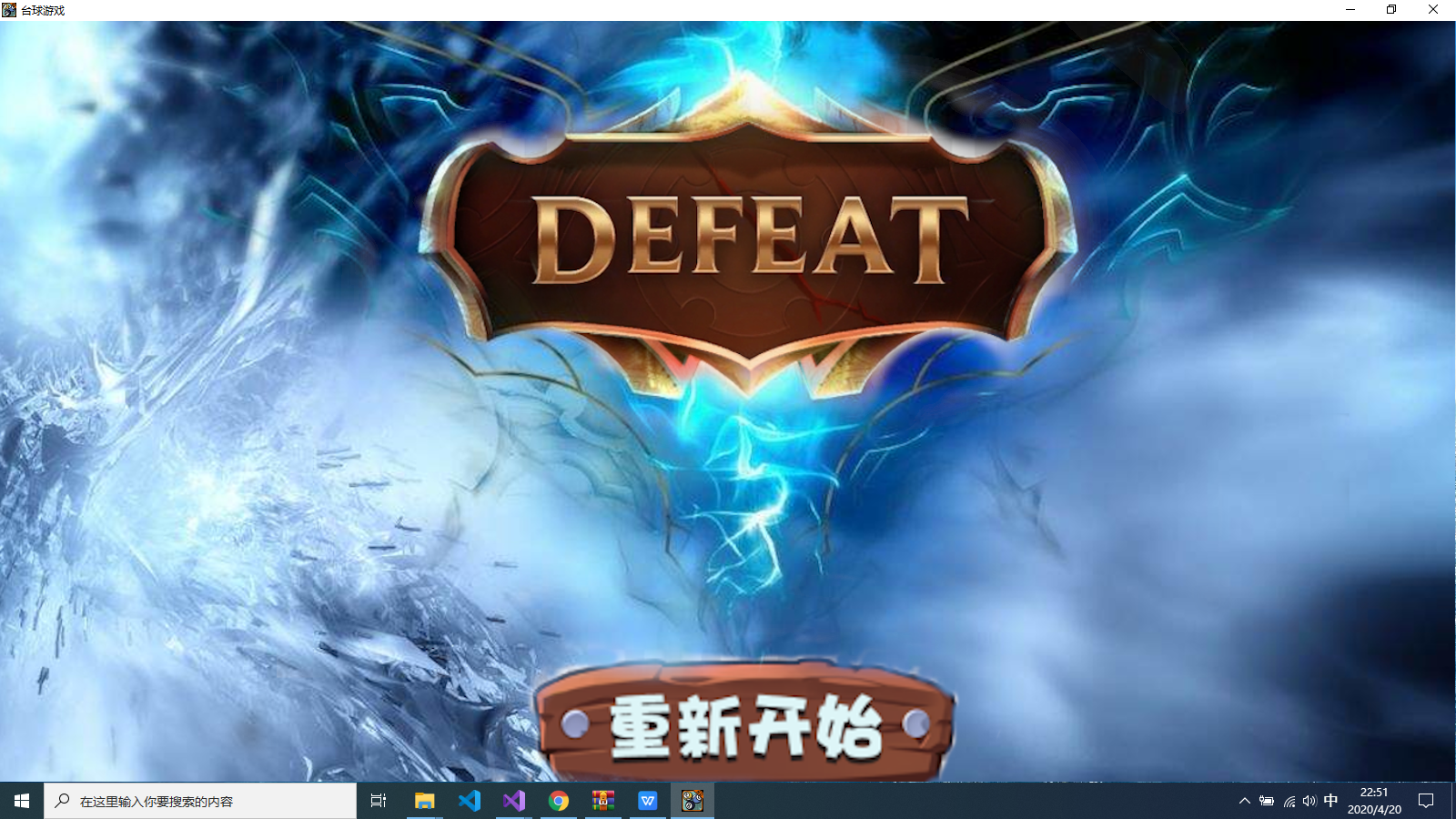
游戏的主要内容是模拟台球的击球和碰撞，此外围绕着台球模拟设计了一些音效，开始界面，成功失败界面等。



进入游戏之后首先展示开始界面，右下角显示到目前的最高记录。当前的最高记录存在文件中。



点击开始游戏之后展示球台和球，台子上共11个球，用其中只有黑球是可以打的球，剩下的红球和绿球是需要打进袋的球。将鼠标放在黑球上向左拉动，松手后球击出。击球的力度和拉的距离正相关，距离越大击球的力度越大。屏幕最上方的进度条显示当前击球的力度，进度条满是最大的力度，大概对应半个屏幕的距离。松手后球会相互碰撞，和台边碰撞或者进袋。这三种情况都会播放音效。球在碰撞台边，互相碰撞或者运动的过程中都会损失能量，反映在程序中是碰撞台边速度降低到80%，相互碰撞下面详细说，正常运行加速度 -100像素/s^2。



当黑球入袋时，判定玩家输，显示失败界面，点击重新开始按钮可以重新开始。

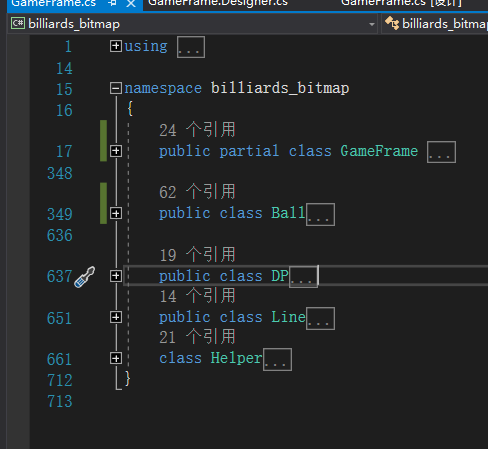


当桌山的所有球都被玩家打进袋后，判定玩家胜利，在打球的过程中统计玩家用了几杆，如果用的杆数比当前的最小值还小，那么判定是一个新纪录，在文件中存下来。

## 项目重点内容：

个人认为自己这个项目做的比较好的地方有两方面，一个是在工程上进行了比较好的面向对象抽象，虽然是第一次写 C#，最后代码量也不小但是复杂度还是很可控的。其次是碰撞的实现，虽然在两个运动的球低速相撞这个场景下还是有问题但是总体上来说效果还是不错的。

在面向对象抽象上，程序中主要包含两个类，一个是GameFrame窗体，一个是球。所有有关球的位置，速度，方向这些变量和移动，碰撞，进袋这些方法都封装到球这个类中。在GameFrame中主要实现图像的绘制，所有的球作为一个List存在GameFrame中。



DP，Line是double类型的点和两个点组成的线段的类，很简单，最后的Helper中主要是计算距离，播放声音，载入图像这种脚手架代码。

关于球的移动和碰撞部分。首先是击球，当GameFrame收到鼠标点击时间时，判断鼠标的位置在哪个球里，并记录当前鼠标位置和球的编号。如果鼠标是点在一个球里，跟踪鼠标的移动，计算鼠标当前位置和按下时的距离，并更新上方的力度显示。当鼠标抬起，计算鼠标拉动的距离和方向，按照拉动方向的反向给球一个初速度，让球开始运动。

之后和台边以及袋子的碰撞都比较简单。球的移动只有两个参数，一个 0~2PI的方向和速度。这个地方因为三角函数学的一般所以感觉代码写的不是很简洁，有很多特殊情况，但是一共也只有四个象限不是很复杂。

比如如果球碰撞台子的上边缘，那么有两种情况，一种是向右上，那么角度应该变成 direction-pi，如果是向左上，那么角度应该变成2pi-direction。和边界的碰撞一共是这8中情况，入袋就是判断和边界碰撞的位置是不是在袋子的范围内。四个底带在实现上实际是四个直角，这个在玩的时候不是很影响但是还是有一些看起来像是进了的球被弹出来。后期准备将这个地方完善成四个斜线，这样应该会更自然。

之后球和球的碰撞，这个因为情况很多，四面八方不可能写case判断。我的总体思路是两个球碰撞的时候，圆心连线的方向上速度互换，在这个基础上可以给一些衰减，垂直圆心的方向上速度不变。在实现上，因为如果用三角函数怕又需要写各种case，所以将两个球的速度先都分成x方向和y方向的分速度，之后这两组分速度投影到圆心连线的方向上，两个速度互换，之后再将两个换过来的速度分解，分别和自己剩下的速度，也就是垂直圆心连线方向的速度合成，最后作为两个球碰撞后的速度，继续运动。这个方法写起来很简单，没有需要特殊判定的情况，而且实际效果上来看大多数场景下是很有效的。

我这发现的一个比较大的问题是当两个很慢的球以一个小角度相撞的时候，这个时候两个球碰撞之后会不断交换速度，速度还会有损失，因此就会粘在一起，甚至是一个球围绕着另一个球转。如果两个球太近就会让绘图进程栈溢出。这个问题个人认为可行的解决方案是在速度很低的情况下降低碰撞计算的频率，这样能避免这种情况。

项目过程中遇到的问题：

1. Double Buffer

刚开始没有意识到DB没有开，图像刷新极慢，整个屏幕都在闪烁。刚开始还以为是显示器或者硬件性能的问题，因为电脑比较老，后来仔细回顾了ppt才想起来是DB的问题。开了双缓存之后偶尔会有掉帧的情况，但是基本只在球速度很快或者有其他球进袋的时候出现。后期针对这点会继续优化。

1. PictureBox

项目开始的时候看了网上的一个flappy bird教程，使用picturebox做的小鸟，因此刚开始做的时候球台和球都是用的pictureBox。PictureBox因为能自己接收点击这些事件，所以代码可以更紧凑一点，而且因为控件内部可以处理绘制所以不需要一直用一个定时器，在一个paint函数中绘图。但是在控件做的球动起来的时候效果十分惨不忍睹，一个球都很不流畅而且后面还拖着尾巴。换到用位图绘制之后完全解决了这个问题。

## 项目体会：

之前基本没有动手写过图形界面的程序，除了命令行的代码基本就是用MVC架构，HTML网页或者引擎去搭建界面。这次实际深入到绘图的层面去编写程序，让我对渲染过程和相关的软硬件有了更深一点的了解。很长时间没写过比较有规模的项目了，码将近1000行代码并看到程序跑起来这种感觉十分开心，好像又回到了大一学数据结构或者高中打竞赛的时候，非常安静和纯粹。

作为一个项目这个游戏还有很多需要优化的地方，下一步会对几个关键的问题逐一做优化。