

Poping Bubbles 小游戏系统设计文档

20231159 赵子聰

一、概述

这是一款运行于 windows 系统的、基于 python（主要为 pygame 模块）的多模式游戏。玩家通过点击屏幕中冒出的泡泡以获取得分。当满足一定条件时游戏结束，玩家可获知自己最终成绩及排名情况。

二、运行环境

玩家须在 windows 64 位操作系统下，保证压缩文件中全部配置文件均存在的情况下运行。

三、功能设计

1. 登录

在开始游戏前，玩家将被要求输入姓名以获取存档内以往记录。

2. 普通模式(normal mode)

在普通模式中，玩家初始生命值为 3，初始 level 值为 1。游戏开始后，有一定数量的泡泡从游戏窗口底端冒出，速度与位置随机。玩家需点击泡泡所在位置来戳破泡泡以获取得分（每点击一个泡泡得 10 分）。泡泡破裂后将消失。若泡泡完好冒出屏幕，则 missed num 将增加 1。玩家每累计获得 200 分，level 值增加 1，同时显示升级动画。Level 值大于 4 时，会随机冒出带有炸弹的泡泡。玩家若点击此类泡泡，炸弹将爆炸，同时玩家生命值减 1。随着 level 值的增加，冒出的泡泡数量将不断增加，速度也逐渐加快。游戏过程中玩家随时都可以按下空格键来暂停、继续游戏。当 missed num 到达 20，或玩家生命值消耗完时，判定玩家失败，游戏结束。以玩家输入的姓名为依据，结束界面将给出历史最好成绩、本次成绩，并显示是否创下新的记录。

3. 限时模式(time limit mode)

在限时模式中，当玩家开始游戏时，系统将开始 30 秒的倒计时。在这 30 秒中，屏幕上将冒出比普通模式更多、更快的泡泡（无炸弹，无等级，不计完好

冒出屏幕的泡泡)。玩家应在这 30 秒内尽可能多地点击泡泡获得得分(每点击一个泡泡得 10 分)。计时结束后,将显示存档中以往限时模式的全部游戏记录的得分前三名的成绩与玩家姓名,以及玩家本次游戏的成绩。

4. 游戏规则介绍

在开始页面玩家还可以选择进入游戏规则介绍界面,其中对两种模式的规则分别进行了介绍。

5. 清除缓存(存档)

玩家可点击开始页面右下角“clear cache”文字以清除缓存。清除后无法恢复。

四、关键算法设计

1. 重叠泡泡的判定及处理

在游戏编写初期,最大的难题便是重叠有多个泡泡时,触发点击事件后,位于所点击区域内的所有泡泡均会爆炸,给玩家带来较差游戏体验。初期该部分算法为当检测到鼠标点击事件后,记录鼠标点击位置,遍历全部泡泡,若点击位置在泡泡的矩形内部,则触发爆炸。经思考并不断改进,目前算法为生成泡泡后,将泡泡对象依次存入列表中。检测到点击事件后,遍历泡泡,若点击位置位于泡泡矩形内,则将其存入一临时列表中。遍历完成后,将 pop 列表中最后一个元素。此算法基于了列表有序这一特点,省去了判断泡泡上下相对位置的步骤。

2. 倒计时器的实现

程序设计过程中最为困难的一个模块就是在限时模式中倒计时器的实现。由于想实现 30 秒倒计时,实时显示剩余秒数的功能,因此较难以程序中的计时变量得以实现,且也难以精准控制时间,后期难以维护。

为了实现此功能,游戏作者在网上搜索了关于多线程的资料,并对 threading 模块中的几种重要方法进行了学习。经过反复试错及可迁移性调试,最终通过另开一个线程用于计时(基于 time 模块),并通过调用全局变量的方式实现了剩余时间值的实时返回。

3. 鼠标无法超出游戏窗口边界的实现

由于游戏窗口宽度较小(600 像素),因此特意设计了阻止鼠标超出游戏窗口边界的功能。该功能通过隐藏鼠标图形、确定鼠标位置、绘制游戏鼠标、判定

是否超出边界并对鼠标绘制位置重定位几个步骤得以实现。

4. 登录模块交互界面的实现

在实现了游戏的基本功能后，游戏作者开始思考如何记录下玩家的历史成绩，并能对最好成绩进行不断更新。为了更好地实现可推广性，游戏作者决定通过玩家在游戏开始时输入姓名，后期通过玩家姓名查找历史记录并更新的效果。

在玩家输入姓名的界面，游戏作者希望能尽可能还原其它应用的用户名输入形式——未进入输入状态并且输入内容为空时，输入框中显示提示字符；鼠标进入输入框后，由指针样变为输入光标；若发生非法输入（输入内容为空或长度过长），则会有相应提示，且提示语会在适当情况下消失。

在前期设计及界面规划后，游戏作者通过定义多个布尔变量来确定输入状态、控制显示内容的方式得以实现。

五、部署与运行

本游戏共分为 11 个界面，每个界面均有对应的特征值，由 `state` 变量及多个 `if...elif...` 语句控制进入。

各界面及对应特征值列表如下：

| 特征值 | 界面名称 |
|-----|----------|
| 0 | 开始界面 |
| 1 | 登录界面 |
| 2 | 模式选择界面 |
| 3 | 说明选择界面 |
| 4 | 普通模式游戏界面 |
| 5 | 普通模式说明界面 |
| 6 | 普通模式结束界面 |
| 7 | 限时模式游戏界面 |
| 8 | 限时模式说明界面 |
| 9 | 限时模式结束界面 |
| 10 | 游戏结束缓冲界面 |

表 1 特征值-界面名称对照表

游戏执行情况如下图所示：

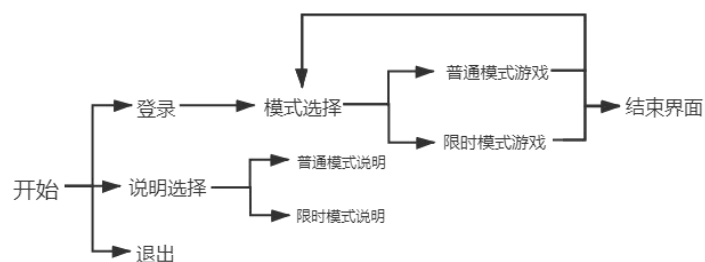


图 1 游戏执行逻辑示意图

六、心得与展望

通过本次课程，我对于 python 这门编程语言有了更加深入的认识与掌握，提升了自学与查找资料的能力，重构了编程思路。

首先是对 Python 的学习。在这个暑假，我第一次接触到了和之前学过的 C 语言差别巨大的这门新的语言——Python。在学习 python 的过程中，我渐渐理解了类与对象、列表元组字典等等完全不同于 C 的数据结构与逻辑，并实实在在体会到了用 python 编程的简洁、准确，而这些都是之前我所掌握的 C 语言所没有的。

其次是我自学与查找资料能力的提升。我从网上找到了 python 的入门视频教程，并完整地学习了全部知识。在编写游戏的过程中，由于涉及到了 pygame、threading 等教程中未讲过的模块，我又从网上查找对应文档与他人经验分享，理解其中我所需方法的参数要求等详细内容。这十几天中，我对于基于 pygame 编程、多线程等有了更深、更全面的了解。

最后是我编程思路的重构。在以前，C 语言课上的程序都只是为了实现较简单功能，代码量也较少。我往往是采取“低头看路，一直往前走”的方式编程，执行遇到错误时再返回去查找问题位置。在此次任务初期，代码量还较少时，我也采取了该方式，还没有遇到什么问题。但随着代码量的不断增加、程序功能的逐渐复杂，我开始使用大量的布尔变量来控制程序运行走向，导致程序运行缓慢、出现了大量效率极低的变量及代码段，程序逻辑也越发模糊。

听了 24 号下午百度章淼老师的讲座后，我重新对这个游戏的执行顺序与逻辑

辑进行了审视。引入了 `state` 变量来标记显示界面，替代了大量复杂的布尔变量，省去了布尔变量的定义、更改状态值等繁琐步骤，也能更清晰地反映程序逻辑。

对于本游戏的展望，目前我有以下设想。第一，除了“水”泡泡之外，随着 `level` 值增加，会出现“木”泡泡（点击一次后变成“水”泡泡）、“石头”泡泡（点击一次后变成“木”泡泡，点击两次后变成“水”泡泡）。第二，可进行联网游戏，排行榜也存储于云端（需检测用户名是否已经存在，防止重名），游戏结束时将显示全网排名（可能涉及到数据库的使用）。第三，开发更多游戏模式，如随机生成特殊道具，拾取道具后择机使用；得分折算成游戏内虚拟金币，增加商城功能，可购买超级道具、升级画面显示效果等。