

FlappyBird

项目大作业报告

手动组队 7

组长：谢东

组员：李诺基，陈建豪

日期：2022.07.08

一、 程序功能介绍

FlappyBird（像素鸟）是一款经典的小游戏，操作非常简单，没有复杂的剧情或是操作技巧，玩家只需通过点击屏幕操纵一只小鸟避开游戏中的绿色管道那样的障碍物即可。如果小鸟碰到管道，则游戏结束。是一个单人，休闲益智游戏。

在打开游戏以后，首先进入游戏的主界面（图 1），在这里玩家可以点击四个按钮，分别对应开始游戏，游戏设置，游戏帮助，退出游戏四个功能按钮。点击不同的按钮即可进入不同的功能界面。同时，鼠标点击到不同的按钮上时，会有字体变红的 hover 效果，使得程序交互性良好。



图 1：游戏主界面

1.1 Play 功能界面

点击 play 按钮以后，进入到游戏的游玩功能，首先一开始是暂停住的，为了防止玩家误按启动了游戏，需要当玩家准备好了以后，先单机点击聚焦于游戏界面，之后按下空格就代表游戏正式开始了（图 2）。



图 2 准备开始游戏界面

玩家操作小鸟用空格键实现跳跃，来躲避场景中出现的管道障碍物。游戏的右上角记录着玩家的得分，每躲避一个管道障碍物，在游戏的右上角就会加 1 分。在游玩过程中，会随着游玩的时间场景的移动速度逐渐加快，但相应的，管道的生成也会变慢一些。同时在游戏中还会随机刷新无敌道具，当玩家吃下无敌道具后，可以一定时间内无视管道碰撞（图 3）。如果玩家碰到管道或者地面，那么游戏即结束，此时可以通过右下角的 Back 按钮回到主菜单选择是否重新开始游戏（图 4）。

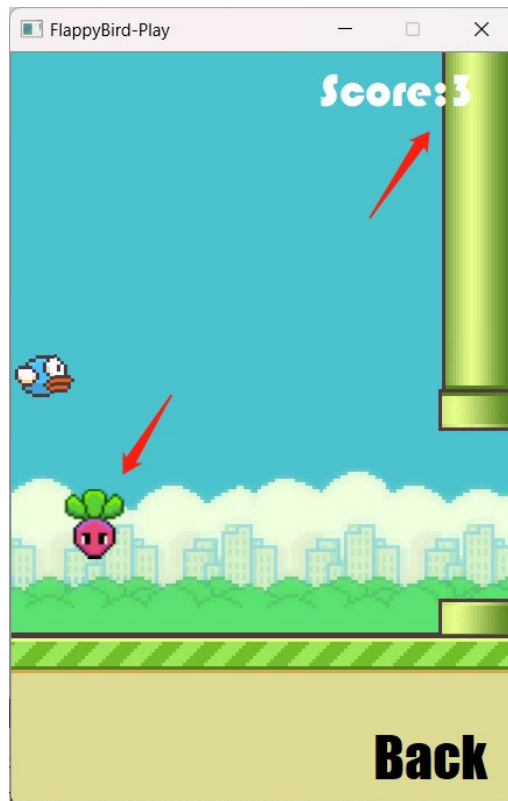


图3 游玩过程中的管道障碍物, 道具, 计分功能



图 4 游戏结束界面，可点击 Back 按钮回单主界面重新开始游戏

1.2 Option 功能界面

Option 功能界面主要实现了对于游戏背景音乐的操作功能。在打开游戏时，游戏会播放轻快的背景音乐。在 Option 界面，首先我们可以通过拖动滑块设置游戏背景音乐的大小。其次，我们可以通过第二行的左右箭头设置音乐是播放上一首还是下一首，而箭头上也设置了当鼠标移动于其上时箭头颜色变化的 hover 效果，提升其交互性。最后可以点击勾选暂停音乐的选项来暂停关闭背景音乐，再次点击即可打开音乐（图 5）。

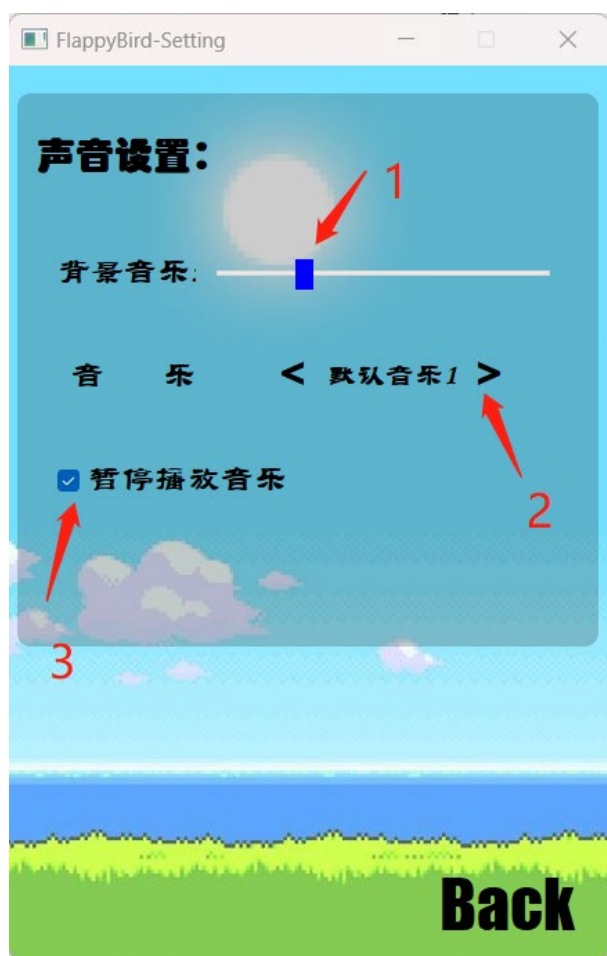


图 5 游戏的设置界面，设置音量，更改音乐，暂停音乐功能实现

1.3 Help 功能界面

点击进入 Help 界面后主要显示游戏的相关玩法介绍和功能介绍。

1.4 Exit 功能按钮

直接退出游戏，关闭程序。

二、项目各模块和类设计细节

项目一共包含九个类模块，其各个类之间的关系如下图（图 6）。

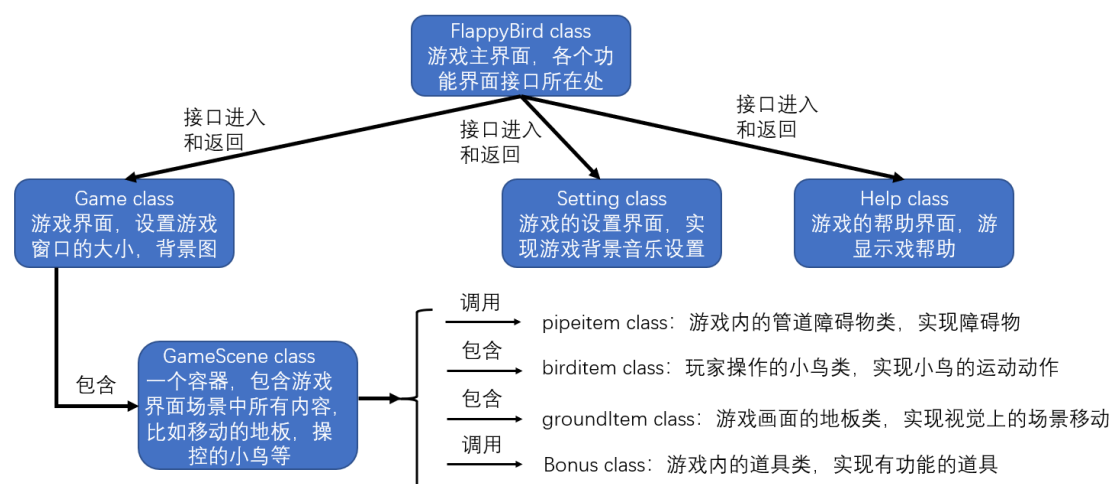


图 6：项目各个类模块关系

每个类的相关设计细节如下：

1. FlappyBird class:

该类作为游戏的主界面，主要实现打开游戏后第一眼看到的界面，包含主题文字，还有跳转各个界面的按钮设计。每一个按钮设置自己的样式并且同时设置按钮跳转界面的槽函数。

2. Game class:

该类是游戏界面，在主界面点击 **play** 按钮后呈现该界面，由该界面设置接下来整个游戏场景大小，以及游戏的背景图片，同时初始化 **GameScene** 类，**GameOver** 类。

3. GameScene class:

该类作为一个容器的作用，在其内实现整个游戏界面内的游玩效果。初始化 **birditem** 对象，实现玩家操作的小鸟在场景中呈现。初始化 **grounditem** 对象，实现地板从左往右移动，使整个场景的呈现出来一种移动效果。此外调用生成 **pipeitem** 对象，通过 **QTime** 类持续为场景内生成管道类障碍物。同理，调用生成 **Bonus** 对象，用 **QTime** 类持续生成道具。同时在整个游戏过程中，检测碰到管道的碰撞信号，以及吃到

道具时的信号，产生相应的功能。

4. pipeitem class:

该类是管道障碍物的定义。通过一张高度大于整个场景的管道图片，我们设置随机数，然后根据得到的随机数设置管道在场景中的位置实现不同管道的生成。同时生成的管道初始化自己的移动动画，从右往左移动。在移动过程中，如果小鸟撞击到管道，则会发出碰撞信号，用于被程序检测做出下一步逻辑。

5. birdItem class:

该类定义玩家操作的小鸟，实现小鸟在游戏中的整个动作过程。包括玩家没有操作时的自然下落，以及玩家在按下空格后，小鸟的跳跃功能的实现。在整个过程中，还实现了小鸟全程扇动翅膀的动画效果。我们分别取小鸟翅膀在上中下的三张贴图，然后在程序运行过程中，我们循环切换三张贴图，这样就实现了小鸟在整个游戏过程中，翅膀扑腾的动画效果。除此之外，小鸟在跳跃，下落时的动画，也同时加上了小鸟旋转的动画，使得看起来更形象一些。

6. groundItem class:

整个其实是一张地面贴图，长度比实际的游戏场景大，我们设置它循环播放从右往左的动画，这样在视觉效果上就是，我们玩家操作的小鸟在一直向右前进的效果。

7. Bonus class:

该类实现的是一个吃到之后就会在一段时间内无视和管道的碰撞，类似于无敌道具。具体实现是通过计时器指定时间间隔，在到时后随机选择是否生成道具，道具的位置也由随机数生成。给道具与管道一样的动画，在碰撞之后给 bird 一个 invincible 的状态，使 bonus 道具消失，并调用一个 QTimer 类实现回到正常状态。在 invincible 的状态下忽略掉碰撞发出的信号即可。

8. Setting class:

该类实现设置界面的所有功能，其中都是对所有设置功能的实现和设计。音量调节的功能通过设置 Horizontal Slider 的槽函数，在我们拖动

的时候，实时获取滑动轴的值，然后设置成为当前播放的背景音乐音量值实现了调节音量的大小。接着实现的背景音乐的切换，也是在左右两个箭头上设置槽函数，在点击的时候，通过参函数切换背景音乐，同时改变两个箭头之间的文本框内的文字，就实现了背景音乐的切换功能。最后的背景音乐暂停，在 `check box` 上设置槽函数，获取它的状态值，通过状态值来变化背景音乐的播放和暂停。其他的就全部是关于这些部件的文本样式设置代码。

9. `Help class`:

该类实现帮助界面的所有功能。就是在跳转这个界面以后，显示关于这个游戏的一些提示信息，每一个功能怎么使用，游戏的玩法。仅仅进行文本显示。

三、小组成员分工情况

小组成员	分工
谢东	游戏主界面向各界面跳转功能，设置界面背景音乐设置功能，帮助界面显示功能，退出按钮功能
李诺基	游玩部分，分数显示相关功能，游戏道具
陈建豪	小鸟、障碍物相关功能，结束界面功能

四、项目总结与反思

在初期进行项目设计的时候，还设计了很多的功能，比如游戏的音效，玩家操作小鸟的换皮肤功能等。但在实际操作的时候，会因为自己开发能力不足，时间来不及，素材不好找等原因，就一一被砍掉了。`QT` 本身的功能还有很多，而现在我们只是因为这个项目迅速的去学习使用一些功能和方法，但本质上对它的理解还十分欠缺，也不了解这个业内开发很常用的框架究竟在业内使

用到哪一种程度了。不过这也本身不是这个课的目的吧。

Qt 整体的语言基于 C++，所以基本的代码可以理解。其最大的特点在于页面的出现和不同页面之间的槽函数连接。因为没有系统性地学过 Qt，所以在实现某些功能时可能用的不是最好的办法，整体效果也不是特别好。但是这份作业让我对一个应用的设计有了个大概的认识，同时也学会了利用 git 等工具来进行代码的传输，从而与队友合作完成这一工程。

在开发过程中，对于一个类，是应该设计成另一个类的成员变量，还是只是在使用的时候，我们才调用生成这个类，关于这个决策有了更深的认识。当涉及变量在另一个类中需要多个函数之间通信的时候，或者是声明一个以后，一直使用到程序结束，那么就应该将这个类设计成另一个类的类成员；而如果只是中间涉及一次，次数少，不用通信的，那么就应该只是在使用时生成实例，用过即销毁。

关于 git 的使用，我们也在小组开发过程中，学会了 git 的一些简单用法。在进行小组团队开发时，发现整合大家的开发进度也是关键的一个环节。在开发时，中间因为 git 还没学会整合时，我们曾采用过手工整合的方式，把别人改动过的代码，阅读后发现和原文件不同的地方手动复制粘贴。但这种整合方式无比低效，而且十分容易出错，并且小组内也不能快速知道整个项目的开发进度。在学习了 git 后，使用它进行项目管理方便太多了，受益匪浅。