

27-作业报告

一、 程序功能介绍

《fruits_cutter》以经典手游《水果忍者》为参考，完成了一个切水果游戏。打开程序，首先会弹出应用图标，点击后会进入主菜单。在主菜单中，玩家可以选择开始游戏，查看本地排行榜或是退出游戏，与此同时，还提供了帮助、背景音乐开关和开发信息按钮，玩家可以在帮助栏查看教程，开关背景音乐，查看开发信息。进入游戏后，屏幕会清空，屏幕下方会不断随机地飞出几种不同的水果。玩家通过按住鼠标左键并滑动鼠标，在屏幕上划出刀光切果水果，每次切到水果便会得分，短时间内切割较多水果还会获得“连斩时刻”奖励，水果移动速度略微放缓（此为特性）。水果之间还会混进炸弹，切到炸弹则炸弹爆炸，游戏结束。除此之外，如果水果未被切割而落出屏幕，玩家会被扣除一点生命值。玩家共有三点生命值，生命值清空游戏也会结束。游戏结束后，系统显示玩家最终得分，并将分数计入排行榜（dojo）。在主界面，玩家可以进入排行榜查看本地历史记录。

二、 项目各模块与类设计细节

刘霄齐：

游戏中的数据检测（health 类与 score 类）：

- 1.在主进程中以图元形式生成，通过重写 paint 函数实现在游戏界面中的显示（分别在屏幕左上角、右上角生成）。
- 2.类中包含反映数据变化的一系列函数，如 dehealth(生命值下降)、checkhealth(如果切到炸弹生命值归零)、checkscore(如果切到水果得分加十)等。
- 3.在游戏进程中，通过将游戏中检测到的切割水果、漏掉水果等信号，由某些中间函数 处理最终与上述函数相连接，并在游戏过程中实时更新画面（update），从而实现游戏 过程中实时的数据显示。

刀光特效的制作（knife 类，游戏中仅实现了最基本的白色刀光）：

- 1.设置两个容器，分别记录鼠标位置（类型 QPoint）与刀光路径（类型 QGraphicsLineItem*）
- 2.读取输入数据（getpoint 函数）：当鼠标左键按下时，读取并记录鼠标位置
- 3.根据记录的位置生成并记录相应的直线图元，并添加最后的 3 个图元到场景中（之前的则删除），从而实现实时的简易拖尾效果
- 4.在游戏进程中，将 getpoint 函数与计时器信号相连，达到在游戏过程中生成拖尾的效果

bgm 的播放与游戏音效的添加：

1. 使用 QMediaPlayer 对象播放 bgm，并通过 setloops (-1) 实现循环播放的效果。
2. 在主页中设置对应按钮，按下后同时实现图标的切换与音乐的播放状态。
3. 使用 QSoundEffect 类实现游戏音效。

胡浩然：

1. 整体运行框架

以 mainmenu 类构建主菜单界面，在其中添加各个功能的按钮（mainmenu 类以 QGraphicsView 为基类，水果形状的按钮基于 QGraphicsItem 实现，帮助和声明按钮基于 QPushButton 实现）。当某个按钮被按下时为对应类分配内存空间，返回主菜单界面时释放对应内存空间。

2. 主游戏运行逻辑

在类 classic 中实现主游戏的运行。classic 类以 QGraphicsView 为基类，classic 对象生成时不断构造出新的 fruit 对象（基于 QTimer 触发），其中每个 fruit 均与 classic 以 connect 函数连接。构造 knife、health、score 等游戏必要元素并与之关联。将 fruit 发出的信号与 health、score 类进行关联以实现加分、降低生命值等功能。当游戏结束时接收 health 发出的信号、终止 fruit 的产生并进行分数结算、回到主界面。

3. 各个类之间的关联

在 mainmenu 类中设置道场类 (dojo)、帮助类 (help)、声明类 (tips) 等的指针并在对应按钮被点击时分配内存空间，在这些类内部存在返回按钮（基于 QPushButton 实现），将按钮与 mainmenu 用关联函数连接，在点击返回按钮时 mainmenu 释放其内存空间。

孙靖皓：

水果相关的代码（fruit 类和 cutted_fruit 类）

1. 与图像相关部分

利用 QPixmap 变量储存了不同种类水果对应的图片，包括了五种水果（完整的和被切开后的两半）以及一种炸弹。

2. 生成部分

在新生成一个水果时，如果是传入缺省参数，则基于随机数随机为水果分配一个种类，并赋予其初始位置、初始速度、初始旋转状态，其所生成的数值均处于在测试后得出的较为合理的区间。同时，还可以手动传入参数，来为水果提供一个确定的初始运动状态，以方便主菜单的搭建。

3. 运动部分

物体在运动过程中，利用 QTimer 进行更新，并通过 setPos、setRotation 等函数实现物体的位置更新。随机生成的水果会在屏幕范围内从下方抛出，在屏幕范围内进行抛体运动。除此之外，还可以手动设置，以使得水果可以固定在某一位置。

4. 切割部分

每一个生成的水果都会基于 QTimer 持续监测是否与按下的鼠标相碰，是否相碰的标准是水果中心与鼠标是否小于一个固定距离。判定被切割后，水果被隐藏，同时在该位置生成两半水果（即两个对应种类的 cutted_fruit 类），向两边炸开，进行平抛运动。

三、 小组成员分工情况

胡浩然：对项目整体进行了规划，整理出整个项目需要完成的各个部分，并对任务进行了划分；编写了程序运行的界面以及程序运作逻辑；完成了 dojo 类（排行榜）、help 类、tips 类等功能。

刘霄齐：负责 score 类、health 类等计数功能；刀光特效的制作；游戏过程中 bgm 与游戏音效的播放。

孙靖皓：主要负责编写所有与水果相关的代码，实现主程序代码中所需要的与水果相关的各种功能。

四、 项目总结与反思

总体来说，项目进行的较为顺利。在项目进行过程中，我们主要遇到了这样的问题：对 QT 语言不熟悉而导致的编写进度缓慢，设计构思与实际代码实现可行性冲突，成员 github 故障导致的代码更新效率低下。不过在初期对项目进行整体规划时，划分的任务相对较为独立，小组成员可以较为方便的进行自己所负责的代码的测试。同时，小组成员多次及时进行线下沟通交流，互通代码想法和意见，很大程度上减少了代码间的冲突。在此后进行类似项目开发时，可从中吸取经验教训，改进开发流程，提高工作效率。