Snake! 设计说明

一、新增功能的介绍

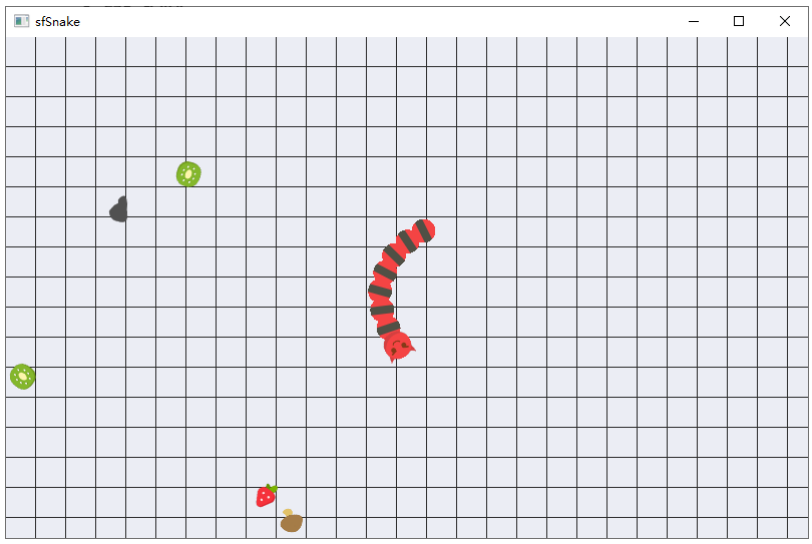
1. 鼠标控制

- 说明

  按照项目要求，实现了**贪吃蛇的鼠标控制。**

  此外，还通过自定义的Button类实现了按键效果，从而实现了**界面的鼠标控制**，至此游戏完全不依赖于键盘输入了。

- 效果



2. 水果设计

- 说明

  按照项目要求，实现了5种不同颜色、得分、出现概率的水果。

  此外，为了提高可玩性，将5种“水果”分别设计为了**草莓、蓝莓、猕猴桃、蘑菇、石头**，分别对应于红色、蓝色、绿色、棕色、黑色。

  另外，为了提高游戏效率，界面中的**水果数目由1个变成5个。**

- 效果

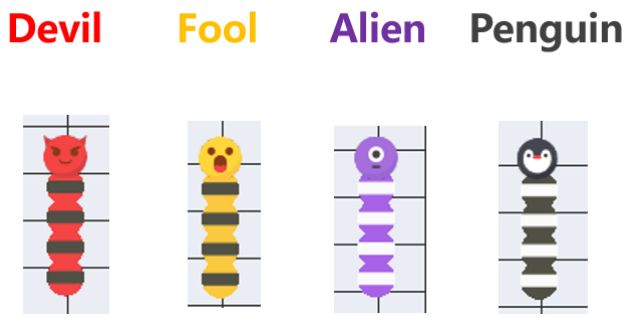
3. 贪吃蛇设计

- 说明

  按照项目要求，实现了精灵版本的贪吃蛇，蛇头用图片，蛇身用圆形叠加矩形绘制。

  此外，为了满足不同玩家的喜好，设计了4种**不同样式的蛇**供玩家在设置界面进行选择。设置界面将在后面说明。

- 效果

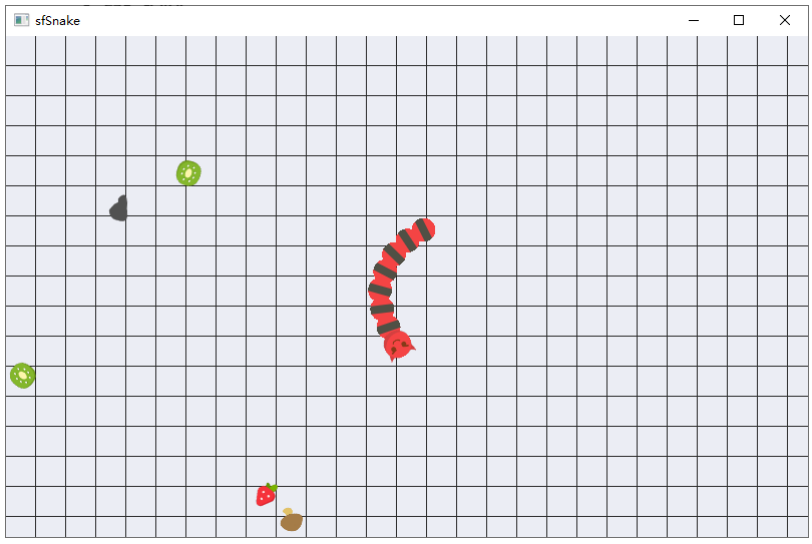


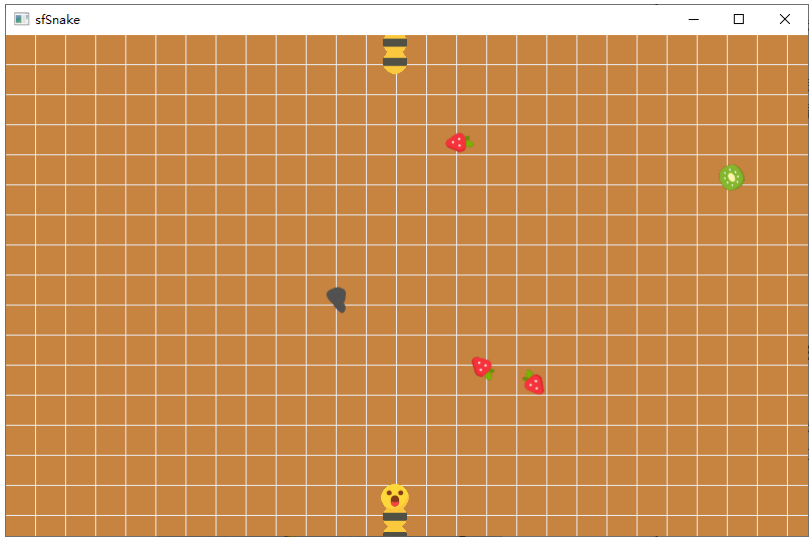
4. 整体界面修改

- 说明

  按照项目要求，实现了背景和网格的白、黑、褐三种颜色的修改。简单起见，这里若背景和网格的颜色相同，则视为关闭显示网格。

- 效果





5. 设置界面与背景音乐

- 说明

  上述的蛇外形选择、背景和网格颜色选择均在设置界面中进行设置。

  此外，还新增了对背景音乐的变更设置，包括原音乐，一共放入**Cute、Pretty、Silent、Default四种不同风格的背景音乐**供玩家选择。音乐是游戏的灵魂！

- 效果

由上至下，分别是对蛇外形、背景、网格、音乐的修改。通过每一项的左右箭头进行切换选择。点击okay后修改才正式生效，点击cancel取消修改。

二、资源管理策略的介绍

1. 实现了统一的资源管理类

为了**防止资源的反复加载**，同时为了**方便资源的维护和扩展**，项目中将所有的纹理(Texture)、字体、声音缓存、背景音乐都统一放在了Recource类里集中管理。同时通过**单例模式**来实现Resource类，使得所有相同的资源只存在一份，从而节省内存空间，实现程序的经济有效。具体代码参见 Resource.h 和 Resource.cpp。

2. 用静态量存放经常用到的映射数据

在程序的运行中，会出现大量类对象的构造和析构，为了节省这些对象的构造开销，代码中将许多经常用到的大数据类型**用static修饰**，从而提升效率。典型的例子是许多存放数据映射的map或vector类型的变量，比如：水果类型到水果分数、纹理名到纹理的映射等等。

三、其他重大改变或优化的列举

1. bug修复

- 将鼠标定位的数值进行随窗口大小的缩放处理，从而修复了窗口放大后鼠标定位错误的bug；
- 水果产生用到的随机生成器改为了通过构造函数传种的方式，以避免反复接种，从而修复了水果不随机的bug；
- 蛇判断出冲突后及时跳出判断的循环结构，以修复由于蛇头与两个相邻结点同时冲突而导致的死亡音效触发两次的bug；
- 为每一个蛇结点、水果等对象设置变换中心，从而修复了左转和右转的角度差异bug；
- 将原版本的Warning报错解决；

2. 设计优化

- 最终得分的计算减去了初始长度，即最低分将是0分；
- 蛇头与蛇身冲突的判断方式改为了蛇头圆心是否在蛇身结点包围盒内的判定方式，变得更加合理了；
- 蛇的增长方式改为了新结点与尾结点重合的方式，变得更加合理了；
- 对水果坐标的随机数范围进行了微调，从而避免水果产生在边界的可能；
- 通过设置音乐循环节的时刻位置实现了新增音乐的无缝衔接循环；
- 为蛇结点、水果对象设置旋转角度，其中蛇结点随着蛇爬行角度进行旋转，水果则以随机旋转角度出现；
- 各种样式美化；

3. 代码优化

- 通过继承的方式简化了许多代码，如蛇身结点类直接继承CircleShape类，蛇头类直接继承Sprite类，而不是像原版本一样包装了一层“空壳”类，然后实现了许多冗余的管道方法；
- 网格也通过单例模式实现，保证网格对象的唯一性；
- 将代码中的许多传值改为了传引用，整型改为了浮点型；

四、项目总结

由于项目内容很有趣所以做得很投入，并且老师的项目要求里也激发了我的许多灵感。当然还有许多更多的灵感没有完全实现，如水果数量的设置、网格间距的设置等等，但由于其它课程压力的原因就暂时做到这样了。值得一提的是，由于C++面向对象的特性，写出来的代码具有很强的可扩展性，如设置界面的每一行设置都是通过一个Setting类实现的，因此若要新增上述没有实现的设置也是比较容易的，再加上统一资源管理的Resource类，代码的可维护性就变得更强了。通过这个项目，我对面向对象的设计方式变得更加的熟悉了。