|  |  |
| --- | --- |
| 图片包含 矢量图形  已生成高可信度的说明  实验报告  ——《青蛙祖玛》小游戏c语言版 | 何勰  2016级信息安全 |

**一、游戏规则**

有一轨道，轨道有两个端点，一个是出发点（左上角），一个是终止点（靠近青蛙为圆形），从轨道一端开始流动宝石（四种颜色），每个宝石都是相邻紧挨的，青蛙处于界面正中，鼠标单击时，青蛙会朝着鼠标所指方向吐出颜色随机的宝石，当相邻宝石颜色相同时，就会消掉，一旦宝石进入终点的洞口则game over。如下图所示：

**二、游戏操作**

点击左键，青蛙处随机产生一个颜色的宝石，可以通过点击左键来改变将要吐出的宝石的颜色，将鼠标移动到要发射的位置，点击右键，青蛙向鼠标点击处吐出宝石。

**三、游戏实现**

**1、模块的划分：**

**①**轨道

**②**轨道上的宝石（即track stone）

**③**青蛙吐出的宝石（即frog stone）

**2、stone结构体**

无论是track stone还是frog stone都是stone，参考飞机大战，设置一个stone的结构体，包含颜色，坐标，被击中的范围，斜率，y方向的步长，状态，类别等属性，其中状态分为hit和move两种，用以消除stone时的判断，类别分为track stone和frog stone两种。

**3、轨道的创建**

数据结构选择链表，根据坐标来创建轨道，选定合适的大小（此代码中设置像素为23）画矩形，即可呈现最终游戏所展现的轨道。

**4、轨道上宝石的创建**

为了实现宝石是从轨道起点处一个一个产生的，在定时器到的时候才调用创建track stone的函数，创建一个宝石，在逻辑上把该宝石插入链表头部（和之后的move函数要匹配，所以此处是头部而不是尾部），然后让宝石初始坐标都是轨道的起点坐标，定时器一到，就把宝石的坐标挪到和轨道相应的下一个，以实现宝石沿着轨道移动，从而不管轨道如何改变，宝石的move函数都不用变，track stone 依旧可以沿轨道移动。需要注意的是，为了实现track stone的颜色在选定的四个颜色中随机生成，用枚举类型来设置颜色，在创建时随机生成0~3的数字，并将该数传递给形参color，在绘图时，用switch case来判断color并用相应颜色的pen和brush来绘图。Frog stone颜色和这个是一样的道理。

**5、青蛙吐出的宝石**

点击左键时，调用创建frog stone的函数，但此时不设置定时器，故我们所看到的宝石不会移动，当点击右键时才设置定时器，然后获取鼠标的坐标（此处遇到的一个问题就是，利用获取鼠标坐标的函数，获取的坐标是整个屏幕上的坐标，所以一定注意要转换成窗口内的坐标，利用ScreenToClient函数），计算斜率，定时器到时调用move函数，就能实现右键指哪儿打哪儿。

**6、定时器的设置**

track stone移动的速度和frog stone移动的速度不一样，故设置两个定时器，来实现速度的不同。程序运行的时候在create track stone 函数里设置track stone的定时器，然后在点击右键时设置frog stone的定时器。

**7、判断是否消除的函数的实现**

参考飞机大战，每一个stone的结构体都有一个x和y的参数，表示击中的范围，然后设置合适的范围（此处是轨道一个矩形大小作为该范围），创建track stone时传递表示范围的参数，每次frog stone移动时都要判断并标记状态，判断是否到了这个范围内，若在其中，则继续判断颜色是否相同，若相同，则标记hit状态，表明是被击中了，此时还没有完，继续遍历该宝石旁边的宝石，若颜色相同，则标记hit。在设置的game paint函数里，判断状态，若是hit则绘制轨道覆盖原来的宝石（即矩形），若是move则绘制宝石（即圆形）。

**四、游戏中遇到的几个重要问题以及最终的解决方法**

**1、内存分配问题**

无论是链表还是game paint，都要注意free和内存释放的问题，一开始我忘记了这一点，发现c盘空间越来越少时才意识到这个问题，虽然电脑重启后会恢复，但在以后编程的时候一定会注意这一点，避免出现内存泄漏的问题。

**2、绘图问题**

最初，我把载入青蛙位图、画轨道、以及画轨道上的石头三部分各写了一个函数，又写了一个game paint的总函数来调用这三个子函数，虽然看起来一目了然，但是发现会有覆盖的问题而且会有延迟导致出现闪动的情况，通过老师的讲解，最终我还是老老实实得写到一个函数里，用同一个dc，根本原因还是在于没有弄清楚载入位图的原理，并且每次重绘时都需要先覆盖掉原来的再绘制，所以画的顺序也需要特别注意，这样每次覆盖才能实现track stone滚动和frog stone发射的效果。

**3、frog stone移动问题**

因为电脑屏幕的坐标和平常所使用的数学坐标是不一样的，斜率计算也不一样，为了要使frog stone正确的移动，我以青蛙所在的位置为中心，把窗口分成了四块分别来考虑斜率的正负。在初次运行时发现有些角度frog stone不能正确的向鼠标所点击的地方发射，最后发现是因为步长设置的过小，导致在计算x方向移动多少时，double向int强制类型转换时零点几的数直接成了0，就导致frog stone只能向垂直方向移动的问题，最后修改了步长，这个问题就基本解决了，但是由于精度的问题，有些地方还是会有一点误差，做不到绝对的精确。