



**数据结构课程设计报告**

**题 目 池塘夜降彩色雨**

**学 院 计算机学院**

**专业班级 网络工程1班**

**学 号 3213006449**

**姓 名 张丽蓉**

**指导教师 杨劲涛**

**2015年 7月 5日**

目录

一、题目 2

[问题描述] 2

[基本要求] 2

二、开发环境 2

三、题目需求分析 3

四、概要设计 3

1、数据类型的定义 3

2、主程序的流程 4

3、各模块之间的调用关系 5

五、详细设计 5

主程序和其他模块的伪码算法 5

六、调试分析 6

1、调试过程遇到的问题及解决 6

2、经验和体会 6

七、用户使用说明 6

附录 9

#### 一、题目

###### [问题描述]

设计一个程序，演示美丽的“池塘夜雨”景色：色彩缤纷的雨点飘飘洒洒地从天而降，滴滴入水有声，溅起圈圈微澜。

###### [基本要求]

（1）雨点的空中出现位置、降落过程的可见程度、入水位置、颜色、最大水圈等等，都是随机确定的；

（2）多个雨点按照各自的随机参数和存在状态，同时演示在屏幕上。

#### 二、开发环境

硬件：PC机

软件：Microsoft Visual Studio 2010

#### 三、题目需求分析

根据题目可知需要设计一个程序，程序运行后，有美丽的池塘夜晚景象，天空飘着彩色的雨，落在池塘时有涟漪产生。而雨点的空中出现位置、入水位置、颜色、最大水圈等等，都是随机确定的。画面中有按钮可以控制雨点数，雨点速度等参数，为了能在画出雨点下落的同时能响应窗口的控件的事件，则需要使用新的线程来画下落的雨点。下落的雨点数可以增减，则需要使用双向的链表来保存雨点的各种状态信息（坐标、大小、角度、涟漪半径等）。每次生成雨点，都在指定的范围内随机出相应的数据，然后每次画完一帧雨点，自增雨点的坐标值或涟漪半径。

#### 四、概要设计

###### 1、数据类型的定义

//雨点类型定义

typedef struct

{

COLORREF color;//雨点颜色

int x,y;//初始坐标

int len;//雨点大小

float angle;//雨点角度

int speed;//落下速度

int thick;//雨点粗细

int radius;//涟漪半径

}droplet;//雨点类型

//雨的双向链表定义

struct dropletchain

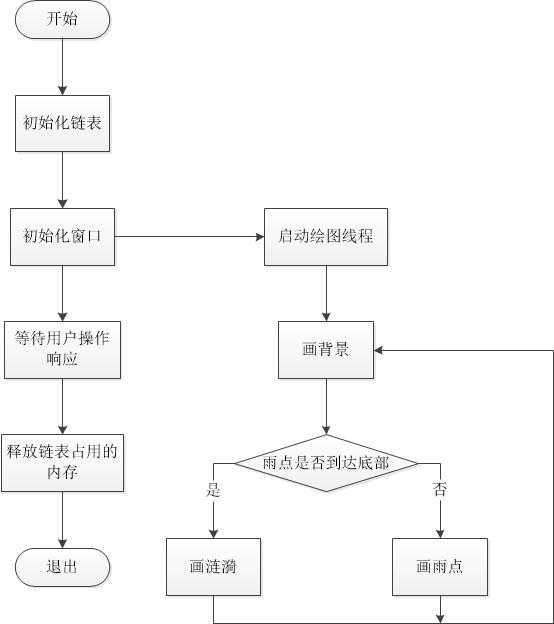
{

droplet \*data;//数据域

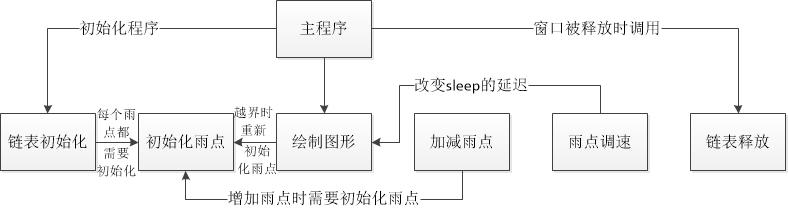
struct dropletchain \*pre,\*next;//指针域

};

###### 2、主程序的流程



###### 3、各模块之间的调用关系



#### 五、详细设计

###### 主程序和其他模块的伪码算法

static UINT onThread(LPVOID wParam)：线程函数

强制转换wParam为当前窗口实例的指针->绘图方法。

void initDroplet(droplet \*droplet)：初始化雨点

随机雨点各个参数

void doThread()

获取画布

while(1)

画背景->读取链表->判断雨点是否到达底部并画雨点或涟漪->自增雨点的坐标或涟漪的半径->sleep延迟

afx\_msg void OnBnClickedButton1()：退出按钮的实现

afx\_msg void OnBnClickedButton2()：暂停按钮

afx\_msg void OnDeltaposSpin(NMHDR \*pNMHDR, LRESULT \*pResult)：雨点数控制按钮

if 增加雨点数

在链表尾部添加雨点->把尾部指针指向新结点->初始化新的雨点

else

if 雨点数剩下一滴

弹出窗口警告

else

删除尾部指向的结点->把尾部指针指向前一个结点

afx\_msg void OnReleasedcaptureSlider(NMHDR \*pNMHDR, LRESULT \*pResult)：雨点速度滑动条

修改窗口成员变量延迟时间的值

#### 六、调试分析

###### 1、调试过程遇到的问题及解决

* 1. 问题：雨点的涟漪使用画圆弧的方法画出来时会带有一根奇怪的线。

解决方法：通过画圆圈的方法画出来，但是却是实心的圆，然后还需要设置内画笔为NULL才可以画空心的圆。

* 1. 问题： 雨点下落及产生涟漪时，经常会跑到画布之外的地方。

解决方法：在产生的涟漪自增其半径后，计算下次绘制涟漪时最大的范围是否会超过画布的范围，若超过了则重新“生成”雨点。

* 1. 问题：静态线程内不能操作窗口。

解决方法：使用适配的方法，把窗口的实例指针传到线程中然后通过新的线程来执行原来实例中的绘图方法即可。

* 1. 问题：初始雨点太少，雨点太短太粗不好看，颜色也太少，还有背景颜色不协调等。

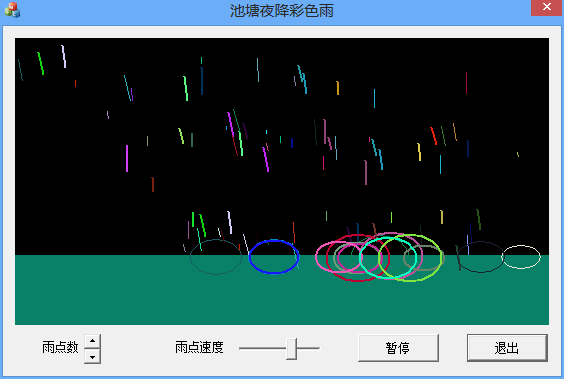
解决方法：修改初始雨点数，雨点线条长度，在电脑的画图里调配颜色再找到比较合适的颜色，修改到最终差不多看着比较协调了。

###### 2、经验和体会

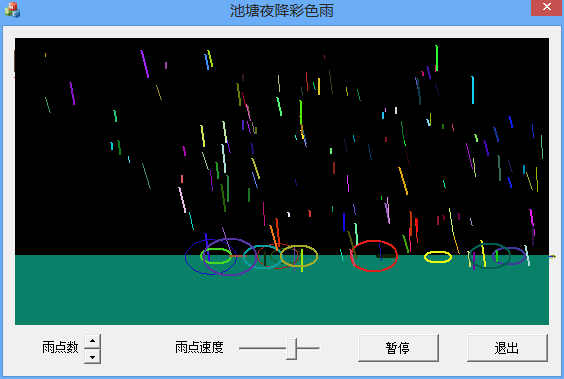
由于vs没有TC中的graphics.h这个图形库,上网查了下，可以使用C++的MFC完成图形的绘制及实现题目要求的下雨的景象，由于没接触过MFC，因而由此简单了解了其使用方法及一些简单的按钮实现功能。经过这个课程设计，大致了解windows的消息处理机制，visual studio的Debug使用。

#### 七、用户使用说明

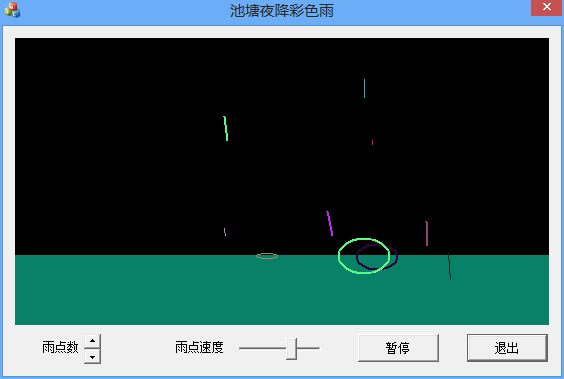
1. 运行程序，可以看到程序的界面如图所示：

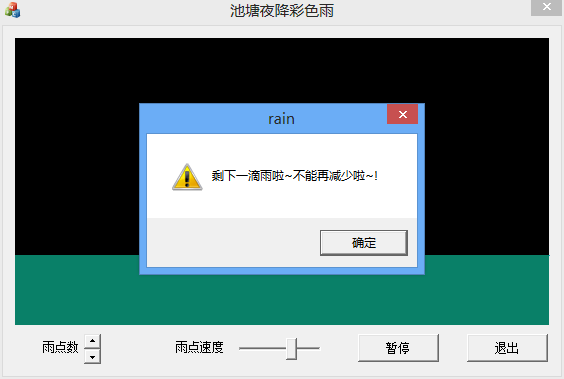


1. 点击雨点数上边的按钮，可以增加雨点数

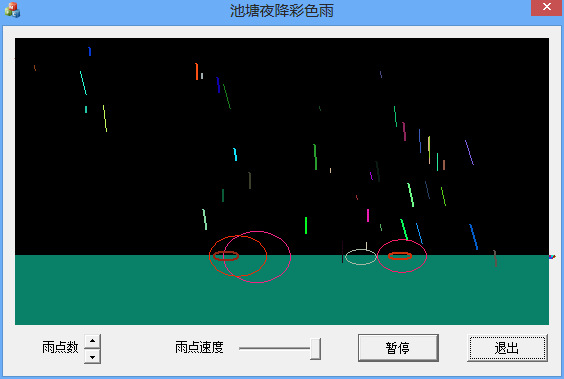


1. 点击雨点数下边的按钮，可以减少雨点数，当雨点数为1时，出现提示，不能继续减少。

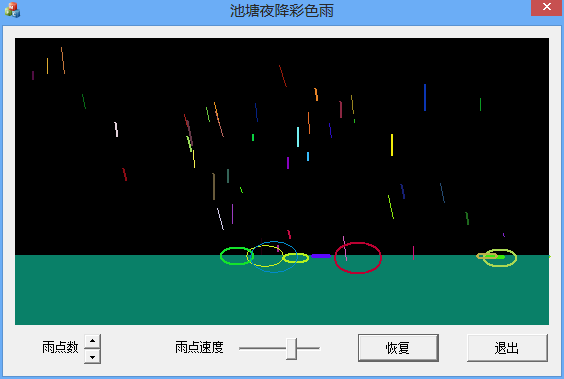




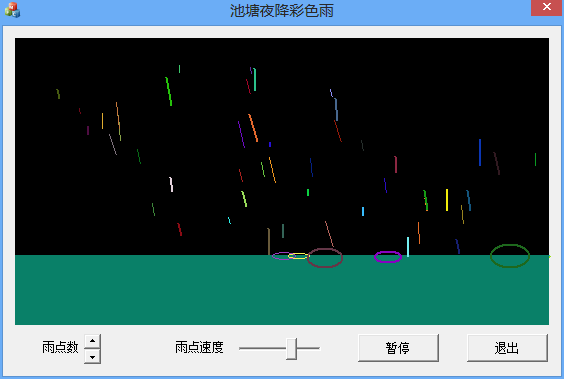
1. 雨点速度滑动条可以控制雨下落的速度，向右则变快，向左则变慢



1. 点击暂停按钮，则整个画面暂停，按钮名称变为“恢复”



1. 点击恢复按钮，则画面开始继续下雨，按钮名称变为“暂停”



1. 点击退出按钮，则退出程序，需重新运行程序才能开始下雨

#### 附录

├─rain.sln 项目文件

├─rain 项目源代码文件夹

│ ├─res 资源文件目录

│ │ ├─rain.ico 项目图标

│ │ ├─rain.rc2 项目资源

│ ├─rain.cpp 项目主文件

│ ├─rain.h 项目头文件

│ ├─rain.rc 项目资源文件

│ ├─rainDlg.cpp 窗口主文件

│ ├─rainDlg.h 窗口头文件

│ ├─resource.h

│ ├─stdafx.cpp

│ ├─stdafx.h

│ ├─targetver.h