

用C写的在桌面上飘雪的特效程序 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/270/2021_2022__E7_94_A8C___E5_86_99_E7_c67_270962.htm #include windows.h

```
#include time.h #include stdlib.h #include iostream.h
const int SnowNumber=500. //雪点数量 struct SnowNode {
POINT postion. //雪点位置 int iColor. //先前的颜色 int iSpeed. //
下落速度 int iMove. //下落距离 int iStick. //粘贴度 }. SnowNode
SnowNodes[SnowNumber]. //雪点数组 int hTimer=0. int
CrWind=0. int CrStep=0. //当前循环步数(用于限速) int
ScreenWidth=0. //屏幕宽度 int ScreenHeight=0. //屏幕高度 void
GetScreenSize(). void CALLBACK TimerProc(HANDLE
hWnd,UINT uMsg,UINT idEvent,DWORD dwTime). void
InitSnowNodes(). void MoveSnowNodes(). int WINAPI
WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
LPSTR lpCmdLine, int nCmdShow ) { MSG msg. //标准windows
消息 LARGE_INTEGER Frequency. //高性能定时器频率
LARGE_INTEGER StartCt,EndCt.//高性能定时器计数 float
ElapsedTime. //时间间隔 srand((unsigned)time(NULL)).
GetScreenSize(). InitSnowNodes().
QueryPerformanceFrequency(amp.StartCt). //执行运算前计数值
if(PeekMessage(amp.EndCt).//执行运算后的计数值
ElapsedTime=(EndCt.QuadPart-StartCt.QuadPart)/Frequency.Qua
dPart. if((ElapsedTime >= 0.0005)) Sleep(2). //简单限速 else
if(ElapsedTime >= 0.0010) Sleep(1). else if(ElapsedTime >= 0.0015)
Sleep(3). } //MessageBox(0,TEXT(“ 消息 “ ),TEXT(“ 消息
```

```

“ ),MB_OK|MB_ICONINFORMATION). return 0. } void
GetScreenSize() {
ScreenWidth=GetSystemMetrics(SM_CXSCREEN).
ScreenHeight=GetSystemMetrics(SM_CYSCREEN). return . } void
CALLBACK TimerProc(HANDLE hWnd,UINT uMsg,UINT
idEvent,DWORD dwTime) { // MessageBox(0,TEXT( “ 消息
“ ),TEXT( “ 消息 “ ),MB_OK|MB_ICONINFORMATION).
srand((unsigned)time(NULL)). if(hTimer==0) {
MessageBox(0,TEXT( “ 创建定时器失败 “ ),TEXT( “ 提示
“ ),MB_OK|MB_ICONINFORMATION). return . }
SetTimer(0,hTimer,((rand() 4)*500),(TIMERPROC)TimerProc).
//// 重设下次风向改变时间 //修改风向 if(CrWind!=0)
CrWind=0. else CrWind=rand()%3-1. return . } void
InitSnowNodes() { HDC hScreenDC=0. int j=0.
hScreenDC=CreateDC( “ DISPLAY “ ,NULL,NULL,NULL).
if(hScreenDC==NULL) { MessageBox(0, “ 获取屏幕DC失败! “ ,
“ 信息 “ ,MB_OK|MB_ICONERROR). return . }
srand((unsigned)time(NULL)). for(j=0;j SnowNumber.j ) {
SnowNodes[j].postion.x=rand()%ScreenWidth.
SnowNodes[j].postion.y=rand()%ScreenHeight.
SnowNodes[j].iColor=GetPixel(hScreenDC,SnowNodes[j].postion
.x,SnowNodes[j].postion.y). SnowNodes[j].iSpeed=(rand()%5 1).
//每次下落距离 ( 1-5 )
SnowNodes[j].iStick=(30-rand()%SnowNodes[j].iSpeed). //粘贴
度 ( 几次循环作一次粘贴连判断 // cout
SnowNodes[j].postion.x “ Y: “

```

```

    SnowNodes[j].postion.y      endl. } DeleteDC(hScreenDC). }
void MoveSnowNodes() { // MessageBox(0,TEXT( “ 消息
    “ ),TEXT( “ 消息 “ ),MB_OK|MB_ICONINFORMATION).
HDC hScreenDC=0. srand((unsigned)time(NULL)). int
x=0,y=0,i=0. hScreenDC=CreateDC( “ DISPLAY
    “ ,NULL,NULL,NULL). if(hScreenDC==NULL) { MessageBox(0,
    “ 获取屏幕DC失败! “ , “ 信息 “ ,MB_OK|MB_ICONERROR).
return . } // TextOut(hScreenDC,0,0, “ 虽然大检查顺顺藤摸瓜克
格勃呀加 “ ,0). for(i=0.i   SnowNumber.i ) { //控制雪点下降速
度 if((CrStep%SnowNodes[i].iSpeed)!=0) continue. //恢复上次被
覆盖点
if((GetPixel(hScreenDC,SnowNodes[i].postion.x,SnowNodes[i].po
stion.y))==0XFFFFFFF)
SetPixel(hScreenDC,SnowNodes[i].postion.x,SnowNodes[i].postio
n.y,SnowNodes[i].iColor). //根据几向作随机飘落
x=SnowNodes[i].postion.x rand()%3 CrWind.
y=SnowNodes[i].postion.y SnowNodes[i].iMove. //积雪 ( 停留 )
效果处理 if( ( (CrStep%SnowNodes[i].iStick)==0) amp.(
(GetPixel(hScreenDC,x,y))!=(GetPixel(hScreenDC,x,y 1))) amp.(
(GetPixel(hScreenDC,x-1,y))!=(GetPixel(hScreenDC,x-1,y 1)))
amp.( (GetPixel(hScreenDC,x 1,y))!=GetPixel(hScreenDC,x 1,y 1))
) { //稍稍调整坐标
if(GetPixel(hScreenDC,x,y-1)==GetPixel(hScreenDC,x,y-2)) { y--.
} else {
if(GetPixel(hScreenDC,x,y-1)==GetPixel(hScreenDC,x,y-2)) y . x
=CrWind. } //画三个雪花点 SetPixel(hScreenDC,x,y,0XFFFFFFF).

```

```
SetPixel(hScreenDC,x 1,y 1,0XFFFFFF). SetPixel(hScreenDC,x-1,y
1,0XFFFFFF). //重生雪点
SnowNodes[i].postion.x=rand()%ScreenWidth.
SnowNodes[i].postion.y=rand().
SnowNodes[i].iColor=GetPixel(hScreenDC,SnowNodes[i].postion
.x,SnowNodes[i].postion.y). } else { if( (x 0) || (x ScreenWidth)
|| (y ScreenHeight)) { SnowNodes[i].postion.x=(rand()).
SnowNodes[i].postion.y=(rand()%ScreenWidth).
SnowNodes[i].iColor=GetPixel(hScreenDC,SnowNodes[i].postion
.x,SnowNodes[i].postion.y). } else { //保存颜色并绘制雪点
SnowNodes[i].iColor=GetPixel(hScreenDC,x,y).
SetPixel(hScreenDC,x,y,0XFFFFFF). //此时保存新雪点位置
SnowNodes[i].postion.x=x. SnowNodes[i].postion.y=y. } } }
DeleteDC(hScreenDC). CrStep . } 100Test 下载频道开通，各类
考试题目直接下载。详细请访问 www.100test.com
```