**โปรแกรมทดสอบกลางภาค**

เขียนโปรแกรมเลียนแบบโปรแกรม Sticky Notes สามารถเพิ่มข้อความได้ สามารถเปลี่ยนสีพื้นหลังได้

**บันทึกผลการทดลอง**

เริ่มจากการประกาศตัวแปรที่ต้องการจะใช้ในส่วนต้นๆของโค้ดโปรแกรม โดยใส่ไว้ใน Global Variables ประกาศฟังก์ชันที่ต้องการใช้ในส่วนของ Foward declarations เริ่มการสร้างหน้าต่าง window ขึ้นมา โดยกำหนดลักษณะของ window ที่ต้องการได้ตามคำสั่ง WNDCLASSEX wcex กำหนดขนาดของ window ที่ถูกเรียกใช้ในส่วน BOOL InitInstance ใช้คำสั่ง hWnd = CreateWindow แล้วตามด้วยขนาดของ window ที่ต้องการ เริ่มเขียนโค้ดในส่วนของการทำงานของโปรแกรมในส่วน LRESULT CALLBACK WndProc เป็นต้นไป ในโปรแกรม Sticky Notes สามารถเพิ่มข้อความได้ เปลี่ยนสีพื้นหลังได้ โดยสามารถเขียนโปรแกรมให้เพิ่มข้อความได้โดยใช้คำสั่ง case WM\_CHAR: จากนั้นเราสร้างแถบเมนูขึ้นมาโดยใช้คำสั่งสร้างในคลาสการสร้าง window คำสั่งที่ใช้ wcex.lpszMenuName = MAKEINTRESOURCE(IDR\_MYMENU); เพื่อใช้เป็นปุ่มตัวเลือกใช้งานเครื่องมือ โดยกำหนดให้มีสองปุ่มแรก คือปุ่ม File กับ Background ในปุ่ม File จะมีตัวเลือกคือ new note และ exit โดยปุ่ม exit คือกดออกจากหน้าต่าง window และปุ่ม Background คือปุ่มเลือกเปลี่ยนสีพื้นหลัง โดยมีสีเขียวให้เลือก และในการกำหนดตัวแปรต่างๆที่สร้างขึ้นมาใหม่ ต้องสร้างไฟล์ .h ไว้เพื่อนิยามให้กับตัวแปรที่กำหนดขึ้นใหม่ และไฟล์ .rc เพื่อเป็นการระบุปุ่มที่ต้องการสร้างขึ้นมาในแถบเครื่องมือ

**Code program**

#include <windows.h>

#include <tchar.h>

#include "resource.h"

// Global Variables:

HINSTANCE hInst; // current instance

TCHAR szTitle[]="Sticky Notes"; // The title bar text

TCHAR szWindowClass[]="WinApp"; // the class name

// Foward declarations of functions included in this code module:

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

int Index=1;

int HIndex = 1;

char H[120] = {0};

int APIENTRY WinMain(HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPSTR lpCmdLine,

int nCmdShow)

{ MSG msg;

MyRegisterClass(hInstance);

if (!InitInstance (hInstance, nCmdShow))

{

return FALSE;

}

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return msg.wParam;

}

ATOM MyRegisterClass(HINSTANCE hInstance)

{

WNDCLASSEX wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

wcex.style = CS\_HREDRAW | CS\_VREDRAW;

wcex.lpfnWndProc = (WNDPROC)WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon (NULL, IDI\_APPLICATION);

wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);

// wcex.lpszMenuName = NULL;

wcex.lpszMenuName = MAKEINTRESOURCE(IDR\_MYMENU);

wcex.lpszClassName = szWindowClass;

wcex.hIconSm = LoadIcon (NULL, IDI\_APPLICATION);

return RegisterClassEx(&wcex);

}

BOOL InitInstance(HINSTANCE hInstance,int nCmdShow)

{

HWND hWnd;

hInst = hInstance; // Store instance handle in our global variable

hWnd = CreateWindow(szWindowClass,szTitle,WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT,0,CW\_USEDEFAULT,0,NULL,NULL,hInstance,NULL);

if (!hWnd)

{

return FALSE;

}

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

return TRUE;

}

LRESULT CALLBACK WndProc(HWND hWnd,UINT message,WPARAM wParam,LPARAM lParam)

{

HBRUSH oldBrush = 0;

HBRUSH newBrush = 0;

bool isChanged = false;

PAINTSTRUCT ps;

HDC hdc;

RECT rt;

HBRUSH hBrush;

char ch[120];

switch (message)

{

case WM\_CHAR:

hdc = GetDC(hWnd);

H[HIndex] =(TCHAR)wParam;;

HIndex++;

TextOut(hdc,10,30,H,120);

ReleaseDC(hWnd, hdc);

break;

case WM\_COMMAND:

switch(LOWORD(wParam))

{

case ID\_FILE\_NEW:

break;

case ID\_FILE\_EXIT:

PostMessage(hWnd, WM\_CLOSE, 0, 0);

break;

case ID\_BG\_G:

case WM\_LBUTTONDOWN:

{

if (!isChanged)

{

newBrush = CreateSolidBrush(RGB(0,255,0)); // green

oldBrush = (HBRUSH)SetClassLong(hWnd,GCL\_HBRBACKGROUND, (LONG)newBrush);

InvalidateRect(hWnd,0,TRUE);

isChanged = true;

}

}

break;

case WM\_RBUTTONDOWN:

{

if (isChanged)

{

SetClassLong(hWnd,GCL\_HBRBACKGROUND,(LONG)oldBrush);

DeleteObject (newBrush);

InvalidateRect(hWnd,0,TRUE);

isChanged = false;

}

}

break;

}

break;

case WM\_CLOSE:

DestroyWindow(hWnd);

break;

case WM\_DESTROY:

if (isChanged)

{

SetClassLong(hWnd,GCL\_HBRBACKGROUND,(LONG)oldBrush);

DeleteObject (newBrush);

isChanged = false;

}

PostQuitMessage(0);

break;

default:

return DefWindowProc(hWnd,message,wParam,lParam);

}

return 0;

}

**ผลการทดลอง**









