ใบงานที่ 2 Message Map

# **วัตถุประสงค์**

1. เพิ่ม Message Map ในโปรแกรม

2. ใช้เครื่องมือต่างๆ ในการสร้าง Application

# **เนื้อหาเบื้องต้น**

https://github.com/Desktop-Programming-Lab-2559/LAB-02/blob/master/LabIntro.md

# **ลำดับการทดลอง**

1. เรียกโปรแกรม Microsoft Visual Studio

2. สร้าง Project ใหม่ โดยเลือกเมนู File >> New >> Project… (Ctrl+Shift+N)

· ช่อง Templates: ให้เลือก Visual C++ และเลือกชนิด project เป็น Empty Project

· ช่อง Name: ให้ใส่ชื่อของ Project เป็น EasyWin32

· ช่อง Location: ให้เลือกตำแหน่งที่จะสร้าง Project (D:\Student\_Code\LAB\_01\_02)

· ส่วนที่เหลือ ให้คงไว้ตามที่ปรากฏ กด OK

3. เพิ่ม source code ให้กับ project โดยการเลือกเมนู PROJECT >> Add New Item…

· ตั้งชื่อไฟล์เป็น EasyMain2.cpp

· กด Add เพื่อเพิ่มไฟล์

4. พิมพ์โปรแกรมดังต่อไปนี้ลงในไฟล์ EasyMain2.cpp

#include <windows.h>

LONG WINAPI WndProc (HWND, UINT, WPARAM, LPARAM);

int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,

LPSTR lpszCmdLine, int nCmdShow)

{

WNDCLASS wc;

HWND hwnd;

MSG msg;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. Define Windows class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

wc.style = 0; // Class style

wc.lpfnWndProc = (WNDPROC) WndProc; // Window procedure address

wc.cbClsExtra = 0; // Class extra bytes

wc.cbWndExtra = 0; // Window extra bytes

wc.hInstance = hInstance; // Instance handle

wc.hIcon = LoadIcon (NULL, IDI\_WINLOGO); // Icon handle

wc.hCursor = LoadCursor (NULL, IDC\_ARROW); // Cursor handle

wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); // Background color

wc.lpszMenuName = NULL; // Menu name

wc.lpszClassName = "MyWndClass"; // WNDCLASS name

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. Register the Windows class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

RegisterClass (&wc);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 3. Create window \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

hwnd = CreateWindow (

"MyWndClass", // WNDCLASS name

"SDK Application", // Window title

WS\_OVERLAPPEDWINDOW, // Window style

CW\_USEDEFAULT, // Horizontal position

CW\_USEDEFAULT, // Vertical position

CW\_USEDEFAULT, // Initial width

CW\_USEDEFAULT, // Initial height

HWND\_DESKTOP, // Handle of parent window

NULL, // Menu handle

hInstance, // Application's instance handle

NULL // Window-creation data

);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 4. Display the window \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ShowWindow (hwnd, nCmdShow);

UpdateWindow (hwnd);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 5. Message loop \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

TranslateMessage (&msg);

DispatchMessage (&msg);

}

return msg.wParam;

}

LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam,

LPARAM lParam)

{

PAINTSTRUCT ps;

HDC hdc;

switch (message) {

case WM\_PAINT:

hdc = BeginPaint (hwnd, &ps);

Ellipse (hdc, 10, 10, 200, 100);

EndPaint (hwnd, &ps);

return 0;

case WM\_DESTROY:

PostQuitMessage (0);

return 0;

}

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

}

4. กดปุ่ม F5 เพื่อดูผลการทำงานของโปรแกรม

5. บันทึกผล

6. ปรับค่า wc (WNDCLASS) มา 5 ค่า

7. บันทึกผล

8. ปรับค่าพารามิเตอร์ของฟังก์ชั่น CreateWindow() มา 5 ค่า

9. บันทึกผล

10. เปลี่ยนฟังก์ชั่น Ellipse() เป็นฟังก์ชันวาดรูป

1. จุด
2. วงกลม
3. สี่เหลี่ยม
4. สามเหลี่ยม
5. เส้นตรง
6. เส้นโค้ง

11. แก้ไขฟังก์ชั่นในการวาดรูป

1. เปลี่ยนสีรูปทรง
2. เปลี่ยนสีเส้น
3. เปลี่ยนลักษณะเส้น
4. เปลี่ยนสีพื้นหลัง

12. เขียนโปรแกรมวาดรูป Doremon

# บันทึกผล

|  |
| --- |
| รูป |
|  |

ตารางการเปลี่ยนค่า WNDCLASS

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); | wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 2); |
| ผลการทดลอง | |
| จากการทดลองได้เปลี่ยนค่า Background color ในคำสัง wc.hbrBackground จาก  wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); ให้เป็น  wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 2); ได้ผลการทดลองคือ สีของ Background เปลี่ยนเป็นสี เทา | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); | wc.hbrBackground = (HBRUSH) (COLOR\_HIGHLIGHT + 1); |
| ผลการทดลอง | |
| จากการทดลองได้เปลี่ยนค่า Background color ในคำสัง wc.hbrBackground จาก  wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); ให้เป็น  wc.hbrBackground = (HBRUSH) (COLOR\_HIGHLIGHT + 1); ได้ผลการทดลองคือ สีของ Background เปลี่ยนเป็นสี น้ำเงิน | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hIcon = LoadIcon (NULL, IDI\_WINLOGO); | wc.hIcon = LoadIcon (NULL, IDI\_ASTERISK); |
| ผลการทดลอง | |
| จากการทดลองได้เปลี่ยน icon ในคำสัง wc.hIcon จาก  wc.hIcon = LoadIcon (NULL, IDI\_WINLOGO); ให้เป็น  wc.hIcon = LoadIcon (NULL, IDI\_ASTERISK); ได้ผลการทดลองคือ ปรากฏ icon INFORMATION อยู่ในแถบมุมซ้ายบน | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hCursor = LoadCursor (NULL, IDC\_ARROW); | wc.hCursor = LoadCursor (NULL, IDC\_NO); |
| ผลการทดลอง | |
| จากการทดลองได้เปลี่ยน Cursor ในคำสัง wc.hCursor จาก  wc.hCursor = LoadCursor (NULL, IDC\_ARROW);ให้เป็น  wc.hCursor = LoadCursor (NULL, IDC\_NO); ได้ผลการทดลองคือ ปรากฏ Cursor เคื่องหมาย ห้าม เมื่อ เลื่อนเมาส์อยู่บน windownfrom | |
| รูป | |
| https://scontent.fbkk1-3.fna.fbcdn.net/v/t35.0-12/21104497_743160919185824_928831143_o.jpg?oh=3cf4429d6f47a7f6ab02be03053c411e&oe=59A14F3D | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.style = 0; | wc.style = CS\_NOCLOSE; |
| ผลการทดลอง | |
| จากการทดลองได้เปลี่ยน คำสัง wc.style จาก  wc.style = 0;ให้เป็น  wc.style = CS\_NOCLOSE;); ได้ผลการทดลองคือ ปรากฏ ไม่สามารถกดปุ่ม close เพื่อปอดโปรแกรมได้ | |
| รูป | |
|  | |

ตารางการเปลี่ยนค่าพารามิเตอร์ของฟังก์ชั่น CreateWindow()

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| "MyWndClass", // WNDCLASS name  "SDK Application", // Window title | "MyWndClass", // WNDCLASS name  "Mr.Jirawat", // Window title |
| ผลการทดลอง | |
| ได้ทำการเปลี่ยนชื่อ title จาก SDK Application เป็น Mr.Jirawat | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| WS\_OVERLAPPEDWINDOW | WS\_VISIBLE, |
| ผลการทดลอง | |
| จากการทดลองได้เปลี่ยน คำสัง WS\_OVERLAPPEDWINDOW ให้เป็น  WS\_VISIBLE, ได้ผลการทดลองคือ ปรากฏ ไม่มีปุ่มกด ปรากฏบน title | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| CW\_USEDEFAULT, // Initial width | 1000, // Initial width |
| ผลการทดลอง | |
| ได้ทำการเปลี่ยนชื่อ // Initial width จาก CW\_USEDEFAULT เป็น 1000  ทำไห้ ได้ดังรูป | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| CW\_USEDEFAULT, // Horizontal position  CW\_USEDEFAULT, // Vertical position  CW\_USEDEFAULT, // Initial width  CW\_USEDEFAULT, // Initial height | 10, // Horizontal position  20, // Vertical position  30, // Initial width  200, // Initial height |
| ผลการทดลอง | |
| ทำให้หน้าต่างโปรแกรมเริ่มต้น เล็กลง ตามขนาดที่เรากำหนด ดังรูป | |
| รูป | |
|  | |
| ค่าเดิม | ค่าใหม่ |
| 500, // Initial width  100, // Initial height | CW\_USEDEFAULT, // Initial width  CW\_USEDEFAULT, // Initial height |
| ผลการทดลอง | |
| ทำให้หน้าต่างโปรแกรมเริ่มต้น มีขนาดตามที่เรากำหนด ดังรูป | |
| รูป | |
|  | |

ตารางการเปลี่ยนฟังก์ชั่นในการวาดรูป

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| Ellipse (hdc, 10, 10, 200, 100); | SetPixel( hdc, 10,30,1);  SetPixel( hdc, 10,32,1); |
| ผลการทดลอง | |
| จะเปลี่ยนจากรูป วงรี เป็นจุด ตามตำแหน่งที่เรากำหนด | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| Ellipse(hdc,10,10,200,100); | Ellipse(hdc,10,10,100,100); |
| ผลการทดลอง | |
| จะเปลี่ยนจากรูป วงรี เป็นจุด ตามตำแหน่งที่เรากำหนด โดยไปเปลี่ยนจุดที่ ตำแหน่งตางๆ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| Ellipse(hdc,10,10,200,100); | Rectangle(hdc, 460, 160, 720, 70); |
| ผลการทดลอง | |
| แสดงตำแหน่ง ของเส้น ตามที่เรากำหนน ในรูปแบบ สี่เหลี่ยม | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| Ellipse(hdc,10,10,200,100); | LineTo(hdc, 50, 150);  LineTo(hdc, 150, 50);  LineTo(hdc, 0, 0); |
| ผลการทดลอง | |
| เป็นการเชื่อมเส้นตรงสามเส้น ให้ต่อกันเป็นรูปสามเหลี่ยมดังรูป | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| Ellipse(hdc,10,10,200,100); | LineTo(hdc, 150, 150); |
| ผลการทดลอง | |
| แสดงตำแหน่ง ของเส้นตรง จากจุด 0.0 ไปยังจุด 150.150 | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| Ellipse(hdc,10,10,200,100); | Arc(hdc,200,200,400,400,200,200,50,1000); |
| ผลการทดลอง | |
| จากการทดลอง ใช้คำส่ง ARC จะทำให้ได้ผลดังรูป | |
| รูป | |
|  | |

ตารางการเปลี่ยนค่าพารามิเตอร์ของฟังก์ชั่นในการวาดรูป

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| Rectangle(hdc, 460, 160, 720, 70); | SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Rectangle(hdc, 460, 160, 720, 70); |
| ผลการทดลอง | |
| จะได้สีพื้นหลังของรูปทรงโดยจะ ผมสมสีกันในแม่สีใน RGB(100,0,50) โดยเลขตัวแรกคือ แดง เหลือง เขียว ตามลำดับ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| LineTo(hdc, 150, 150); | SelectObject(hdc,CreatePen(PS\_SOLID,5,RGB(255,255,0)));  LineTo(hdc, 150, 150); |
| ผลการทดลอง | |
| จะได้เส้นสีเหลือง และเพิ่มขนาดเส้นขึ้นมา | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| LineTo(hdc, 150, 150); | SelectObject(hdc,CreatePen(PS\_DOT,1,RGB(255,0,0)));  LineTo(hdc, 150, 150); |
| ผลการทดลอง | |
| เปลี่ยนสีเส้นตรงเป็นสีแดง และเปลี่ยนลักษณะเส้นตรง เป็นเส้นประ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); | wc.hbrBackground = (HBRUSH) (COLOR\_HIGHLIGHT + 1); |
| ผลการทดลอง | |
| จากการทดลองได้เปลี่ยนค่า Background color ในคำสัง wc.hbrBackground จาก  wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); ให้เป็น  wc.hbrBackground = (HBRUSH) (COLOR\_HIGHLIGHT + 1); ได้ผลการทดลองคือ สีของ Background เปลี่ยนเป็นสี น้ำเงิน | |
| รูป | |
|  | |

โปรแกรมวาดรูป Doremon

|  |
| --- |
| โปรแกรม |
| #include<windows.h>  LONG WINAPI WndProc(HWND,UINT,WPARAM,LPARAM);  int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpszCmdLine,int nCmdShow)  {  WNDCLASS wc;  HWND hwnd;  MSG msg;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. Define Windows class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  wc.style = 0; // Class style  wc.lpfnWndProc = (WNDPROC) WndProc; // Window procedure address  wc.cbClsExtra = 0; // Class extra bytes  wc.cbWndExtra = 0; // Window extra bytes  wc.hInstance = hInstance; // Instance handle  wc.hIcon = LoadIcon (NULL, IDI\_WINLOGO); // Icon handle  wc.hCursor = LoadCursor (NULL, IDC\_ARROW); // Cursor handle  wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); // Background color  wc.lpszMenuName = NULL; // Menu name  wc.lpszClassName = "MyWndClass"; // WNDCLASS name  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. Register the Windows class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  RegisterClass(&wc);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 3. Create window \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  hwnd = CreateWindow (  "MyWndClass", // WNDCLASS name  "SDK Application", // Window title  WS\_OVERLAPPEDWINDOW, // Window style  CW\_USEDEFAULT, // Horizontal position  CW\_USEDEFAULT, // Vertical position  CW\_USEDEFAULT, // Initial width  CW\_USEDEFAULT, // Initial height  HWND\_DESKTOP, // Handle of parent window  NULL, // Menu handle  hInstance, // Application's instance handle  NULL // Window-creation data  );  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 4. Display the window \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  ShowWindow (hwnd, nCmdShow);  UpdateWindow (hwnd);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 5. Message loop \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  while(GetMessage(&msg,NULL,0,0)){  TranslateMessage(&msg);  DispatchMessage(&msg);  }  return msg.wParam;  }  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam,LPARAM lParam)  {  PAINTSTRUCT ps;  HDC hdc;  switch(message){  case WM\_PAINT:  hdc = BeginPaint(hwnd,&ps);    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no.1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //kha    SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,115,190,345,400);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 100, 270,360, 400);    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 160, 280,225, 385);  Rectangle(hdc, 230, 280,298, 385);  Ellipse(hdc,150,190,310,350);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(66,143,232)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 161, 300,297,349);  Rectangle(hdc, 290, 220,310, 240);  Rectangle(hdc, 153, 220,180, 240);  Rectangle(hdc, 170, 205,290, 220);  Rectangle(hdc, 165, 215,295, 225);  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,227,360,305,392);  Ellipse(hdc,153,360,229,392);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  //thong  Ellipse(hdc,160,330,300,195); //wi tong  Ellipse(hdc,180,300,280,210); //wi tong  Rectangle(hdc, 181, 260,279, 261);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Rectangle(hdc, 180, 221,282, 260);    LineTo(hdc, 163, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,105,260,155,305);  Ellipse(hdc,305,260,355,305);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));      //head    Ellipse(hdc,125,210,335,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,125,209,335,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,145,210,315,70); //small h  Ellipse(hdc,192,100,229,60); //ta  Ellipse(hdc,231,100,266,60); //ta  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Ellipse(hdc,219,118,239,98); //jamug  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Rectangle(hdc, 170, 195,289, 205); //pogkor  Ellipse(hdc,167,195,174,205); //jamug  Ellipse(hdc,285,195,292,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,0,0)));  Ellipse(hdc,168,195,175,205); //jamug  Ellipse(hdc,284,195,291,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 198, 205,262, 260);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SelectObject(hdc,CreateSolidBrush(RGB(255,255,0)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,215,190,245,220);//kading  Ellipse(hdc,213,197,247,205);  Rectangle(hdc, 230, 205,231,220);  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,225,206,235,214);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,207,85,220,70); // ta b r  Ellipse(hdc,238,85,250,70); // ta b R  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,170,145,290,170); // m  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Ellipse(hdc,170,130,290,168); //w pagbon    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  Rectangle(hdc, 229, 118,230, 170); //s jamug      //nuad  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 218, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 215, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 155, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 217, 141);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 160, 147);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 241, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 300, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 240, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 303, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 240, 140);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 297, 143);    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no.2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    EndPaint(hwnd,&ps);  return 0;  case WM\_DESTROY:  PostQuitMessage(0);  return 0;  }  return DefWindowProc (hwnd, message, wParam, lParam);  } |
| ผลการทดลอง |
| จากการสร้างวัตถุทรงกลมละสี่เหลี่ยม มาทำเป็นรูปโดเรม่อน ทำให้ทราบว่า วัตถุชิ้นแรกจะโดนวัตถุชิ้นถัดมาทับอยู่ด้านบน |
| รูป |
|  |

# สรุปผลการทดลอง

# คำถามท้ายการทดลอง

1. อธิบายการทำงานของโปรแกรมในข้อ 4

ตอบ จะทำการสร้างหน้าต่างขึ้นมา โดยขนาดของหน้าต่างเราสามารถปรับแต่งได้ที่ ชุดคำสั่ง Create windowอโดยเราสามารถ เปลี่ยนชื่อของ titlte bar ได้ที่ชุดคำสั่งนี้

ส่วนการทำงานของชุดคำสั่ง Define Windows class จะทำการแก้ใขรูปแบบ titlte bar ให้มี ไอคอนต่างๆหรือไม่มีก็ได้

การทำงานของชุดคำสั่ง Message loop คือการสร้าง ข้อมูลต่างที่เราต้องการในตัว body ของหน้าต่างที่เรากำหนด อาทิ รูปทรงต่างๆเป็นต้น

1. บอกพารามิเตอร์ของ WNDCLASS

COLOR\_ACTIVEBORDER  
COLOR\_ACTIVECAPTION  
COLOR\_APPWORKSPACE  
COLOR\_BACKGROUND  
COLOR\_BTNFACE  
COLOR\_BTNSHADOW  
COLOR\_BTNTEXT  
COLOR\_CAPTIONTEXT  
COLOR\_GRAYTEXT  
COLOR\_HIGHLIGHT  
COLOR\_HIGHLIGHTTEXT  
COLOR\_INACTIVEBORDER  
COLOR\_INACTIVECAPTION  
COLOR\_MENU  
COLOR\_MENUTEXT  
COLOR\_SCROLLBAR  
COLOR\_WINDOW  
COLOR\_WINDOWFRAME  
COLOR\_WINDOWTEXT

1. บอกพารามิเตอร์ และหน้าที่ของฟังก์ชั่น CreateWidow

Button

**COMBOBOX**

**EDIT**

**LISTBOX**

**MDICLIENT**

**RICHEDIT\_CLASS**

**SCROLLBAR**

**STATIC**

**หน้าที่ของ** ฟังก์ชั่น CreateWidow คือ ออกแบบหน้าต่างของโปรแกรม ที่เราสร้างขึ้น ตั่งชื่อ หน้าต่างของเรา สร้าง ไอคอน ฯ

1. บอกหน้าที่ของฟังก์ชั่น RegisterClass(), ShowWindow(), UpdateWindow(), TranslateMessage(), DispatchMessage()

RegisterClass ทำหน้าที่เป็นตัวชี้ไปยัง [**WNDCLASS**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms633576(v=vs.85).aspx)โครงสร้าง ต้องกรอกข้อมูลโครงสร้างด้วยแอตทริบิวต์คลาสที่เหมาะสมก่อนส่งผ่านไปยังฟังก์ชันได้

ShowWindowควบคุมหน้าต่างที่จะแสดงผล

UpdateWindow ทำการอัพเดทหน้าต่างที่จะได้รับการปรับปรุง

TranslateMessage แปลข้อความเสมือนเป็นข้อความตัวอักษร

DispatchMessage ส่งข้อความไปยังหน้าต่างจักการข้อความ

1. บอกชื่อและหน้าที่ WM\_MESSAGE มา 10 ชื่อ

WM\_PAINT ใช้วาดรูป

WM\_NULL ไม่โชว์ข้อมูล

WM\_DESTROY เป็นคำสั่งจบการทำงานของโปรแกรม

WM\_GETFONT ดึงข้อมูลแบบอักษรที่ตัวควบคุมกำลังวาดข้อความอยู่

WM\_SIZING ปรับขนาดของหน้าต่าง

WM\_STYLECHANGING style ของหน้าต่างหรือลักษณะหน้าต่างแบบขยายกำลังเปลี่ยนแปลงอยู่ หรือไม่  
 WM\_SETTEXT ตั้งค่าข้อความของหน้าต่าง

WM\_SETICON ตั้งค่าไอคอนของหน้าต่าง

WM\_SETFONT ตั้งค่าแบบอักษรที่ตัวควบคุมจะใช้เมื่อวาดข้อความ  
 WM\_GETTEXT คัดลอกข้อความที่ตรงกับหน้าต่างในบัฟเฟอร์ที่ผู้โทรเรียก

1. เขียนโปรแกรมสร้าง Doraemon จำนวน 5 ตัว

|  |
| --- |
| โปรแกรม |
| #include<windows.h>  LONG WINAPI WndProc(HWND,UINT,WPARAM,LPARAM);  int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpszCmdLine,int nCmdShow)  {  WNDCLASS wc;  HWND hwnd;  MSG msg;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. Define Windows class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  wc.style = 0; // Class style  wc.lpfnWndProc = (WNDPROC) WndProc; // Window procedure address  wc.cbClsExtra = 0; // Class extra bytes  wc.cbWndExtra = 0; // Window extra bytes  wc.hInstance = hInstance; // Instance handle  wc.hIcon = LoadIcon (NULL, IDI\_WINLOGO); // Icon handle  wc.hCursor = LoadCursor (NULL, IDC\_ARROW); // Cursor handle  wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); // Background color  wc.lpszMenuName = NULL; // Menu name  wc.lpszClassName = "MyWndClass"; // WNDCLASS name  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. Register the Windows class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  RegisterClass(&wc);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 3. Create window \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  hwnd = CreateWindow (  "MyWndClass", // WNDCLASS name  "SDK Application", // Window title  WS\_OVERLAPPEDWINDOW, // Window style  CW\_USEDEFAULT, // Horizontal position  CW\_USEDEFAULT, // Vertical position  CW\_USEDEFAULT, // Initial width  CW\_USEDEFAULT, // Initial height  HWND\_DESKTOP, // Handle of parent window  NULL, // Menu handle  hInstance, // Application's instance handle  NULL // Window-creation data  );  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 4. Display the window \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  ShowWindow (hwnd, nCmdShow);  UpdateWindow (hwnd);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 5. Message loop \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  while(GetMessage(&msg,NULL,0,0)){  TranslateMessage(&msg);  DispatchMessage(&msg);  }  return msg.wParam;  }  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam,LPARAM lParam)  {  PAINTSTRUCT ps;  HDC hdc;  switch(message){  case WM\_PAINT:  hdc = BeginPaint(hwnd,&ps);    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no.1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //kha    SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,115,190,345,400);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 100, 270,360, 400);    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 160, 280,225, 385);  Rectangle(hdc, 230, 280,298, 385);  Ellipse(hdc,150,190,310,350);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(66,143,232)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 161, 300,297,349);  Rectangle(hdc, 290, 220,310, 240);  Rectangle(hdc, 153, 220,180, 240);  Rectangle(hdc, 170, 205,290, 220);  Rectangle(hdc, 165, 215,295, 225);  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,227,360,305,392);  Ellipse(hdc,153,360,229,392);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  //thong  Ellipse(hdc,160,330,300,195); //wi tong  Ellipse(hdc,180,300,280,210); //wi tong  Rectangle(hdc, 181, 260,279, 261);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Rectangle(hdc, 180, 221,282, 260);    LineTo(hdc, 163, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,105,260,155,305);  Ellipse(hdc,305,260,355,305);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));      //head    Ellipse(hdc,125,210,335,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,125,209,335,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,145,210,315,70); //small h  Ellipse(hdc,192,100,229,60); //ta  Ellipse(hdc,231,100,266,60); //ta  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Ellipse(hdc,219,118,239,98); //jamug  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Rectangle(hdc, 170, 195,289, 205); //pogkor  Ellipse(hdc,167,195,174,205); //jamug  Ellipse(hdc,285,195,292,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,0,0)));  Ellipse(hdc,168,195,175,205); //jamug  Ellipse(hdc,284,195,291,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 198, 205,262, 260);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SelectObject(hdc,CreateSolidBrush(RGB(255,255,0)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,215,190,245,220);//kading  Ellipse(hdc,213,197,247,205);  Rectangle(hdc, 230, 205,231,220);  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,225,206,235,214);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,207,85,220,70); // ta b r  Ellipse(hdc,238,85,250,70); // ta b R  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,170,145,290,170); // m  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Ellipse(hdc,170,130,290,168); //w pagbon    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  Rectangle(hdc, 229, 118,230, 170); //s jamug      //nuad  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 218, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 215, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 155, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 217, 141);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 160, 147);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 241, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 300, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 240, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 303, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 240, 140);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 297, 143);    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no.2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //kha    SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,365,190,595,400);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 350, 270,610, 400);    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 410, 280,505, 385);  Rectangle(hdc, 480, 280,548, 385);  Ellipse(hdc,400,190,560,350);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(66,143,232)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 411, 300,547,349);  Rectangle(hdc, 540, 220,560, 240);  Rectangle(hdc, 403, 220,430, 240);  Rectangle(hdc, 420, 205,540, 220);  Rectangle(hdc, 415, 215,545, 225);  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,477,360,555,392);  Ellipse(hdc,403,360,479,392);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  //thong  Ellipse(hdc,410,330,550,195); //wi tong  Ellipse(hdc,430,300,530,210); //wi tong  Rectangle(hdc, 431, 260,529, 261);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Rectangle(hdc, 430, 221,532, 260);    LineTo(hdc, 413, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,355,260,405,305);  Ellipse(hdc,555,260,605,305);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));      //head    Ellipse(hdc,375,210,585,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,375,209,585,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,395,210,565,70); //small h  Ellipse(hdc,445,100,479,60); //ta  Ellipse(hdc,481,100,516,60); //ta  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Ellipse(hdc,469,118,489,98); //jamug  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Rectangle(hdc, 420, 195,539, 205); //pogkor  Ellipse(hdc,417,195,424,205); //jamug  Ellipse(hdc,535,195,542,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,0,0)));  Ellipse(hdc,418,195,425,205); //jamug  Ellipse(hdc,534,195,541,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 448, 205,512, 260);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SelectObject(hdc,CreateSolidBrush(RGB(255,255,0)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,465,190,495,220);//kading  Ellipse(hdc,463,197,497,205);  Rectangle(hdc, 480, 205,481,220);  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,475,206,485,214);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,457,85,470,70); // ta b r  Ellipse(hdc,488,85,500,70); // ta b R  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,420,145,540,170); // m  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Ellipse(hdc,420,130,540,168); //w pagbon    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  Rectangle(hdc, 479, 118,480, 170); //s jamug      //nuad  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 468, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 465, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 405, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 467, 141);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 410, 147);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 491, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 550, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 490, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 553, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 490, 140);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 547, 143);  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no.3\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //kha    SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,615,190,845,400);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 600, 270,860, 400);    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 660, 280,725, 385);  Rectangle(hdc, 730, 280,798, 385);  Ellipse(hdc,650,190,810,350);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(66,143,232)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 661, 300,797,349);  Rectangle(hdc, 790, 220,810, 240);  Rectangle(hdc, 653, 220,680, 240);  Rectangle(hdc, 670, 205,790, 220);  Rectangle(hdc, 665, 215,795, 225);  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,727,360,805,392);  Ellipse(hdc,653,360,729,392);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  //thong  Ellipse(hdc,660,330,800,195); //wi tong  Ellipse(hdc,680,300,780,210); //wi tong  Rectangle(hdc, 681, 260,779, 261);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Rectangle(hdc, 680, 221,782, 260);    LineTo(hdc, 663, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,605,260,655,305);  Ellipse(hdc,805,260,855,305);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));      //head    Ellipse(hdc,625,210,835,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,625,209,835,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,645,210,815,70); //small h  Ellipse(hdc,692,100,729,60); //ta  Ellipse(hdc,731,100,766,60); //ta  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Ellipse(hdc,719,118,739,98); //jamug  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Rectangle(hdc, 670, 195,789, 205); //pogkor  Ellipse(hdc,667,195,674,205); //jamug  Ellipse(hdc,785,195,792,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,0,0)));  Ellipse(hdc,668,195,675,205); //jamug  Ellipse(hdc,784,195,791,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 698, 205,762, 260);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SelectObject(hdc,CreateSolidBrush(RGB(255,255,0)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,715,190,745,220);//kading  Ellipse(hdc,713,197,747,205);  Rectangle(hdc, 730, 205,731,220);  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,725,206,735,214);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,707,85,720,70); // ta b r  Ellipse(hdc,738,85,750,70); // ta b R  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,670,145,790,170); // m  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Ellipse(hdc,670,130,790,168); //w pagbon    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  Rectangle(hdc, 729, 118,730, 170); //s jamug      //nuad  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 718, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 715, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 655, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 717, 141);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 660, 147);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 741, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 800, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 740, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 803, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 740, 140);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 797, 143);  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no.4\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,865,190,1095,400);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 850, 270,1110, 400);    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 910, 280,1005, 385);  Rectangle(hdc, 980, 280,1048, 385);  Ellipse(hdc,900,190,1060,350);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(66,143,232)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 911, 300,1047,349);  Rectangle(hdc, 1040, 220,1060, 240);  Rectangle(hdc, 903, 220,930, 240);  Rectangle(hdc, 920, 205,1040, 220);  Rectangle(hdc, 915, 215,1045, 225);  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,977,360,1055,392);  Ellipse(hdc,903,360,979,392);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  //thong  Ellipse(hdc,910,330,1050,195); //wi tong  Ellipse(hdc,930,300,1030,210); //wi tong  Rectangle(hdc, 931, 260,1029, 261);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Rectangle(hdc, 930, 221,1032, 260);    LineTo(hdc, 913, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,855,260,905,305);  Ellipse(hdc,1055,260,1105,305);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));      //head    Ellipse(hdc,875,210,1085,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,875,209,1085,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,895,210,1065,70); //small h  Ellipse(hdc,945,100,979,60); //ta  Ellipse(hdc,981,100,1016,60); //ta  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Ellipse(hdc,969,118,989,98); //jamug  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Rectangle(hdc, 920, 195,1039, 205); //pogkor  Ellipse(hdc,917,195,924,205); //jamug  Ellipse(hdc,1035,195,1042,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,0,0)));  Ellipse(hdc,918,195,925,205); //jamug  Ellipse(hdc,1034,195,1041,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 948, 205,1012, 260);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SelectObject(hdc,CreateSolidBrush(RGB(255,255,0)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,965,190,995,220);//kading  Ellipse(hdc,963,197,997,205);  Rectangle(hdc, 980, 205,981,220);  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,975,206,985,214);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,957,85,970,70); // ta b r  Ellipse(hdc,988,85,1000,70); // ta b R  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,920,145,1040,170); // m  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Ellipse(hdc,920,130,1040,168); //w pagbon    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  Rectangle(hdc, 979, 118,980, 170); //s jamug      //nuad  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 968, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 965, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 905, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 967, 141);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 910, 147);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 991, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1050, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 990, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1053, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 990, 140);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1047, 143);    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no.5\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //kha    SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,1115,190,1345,400);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 1100, 270,1360, 400);    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 1160, 280,1225, 385);  Rectangle(hdc, 1230, 280,1298, 385);  Ellipse(hdc,1150,190,1310,350);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(66,143,232)));  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Rectangle(hdc, 1161, 300,1297,349);  Rectangle(hdc, 1290, 220,1310, 240);  Rectangle(hdc, 1153, 220,1180, 240);  Rectangle(hdc, 1170, 205,1290, 220);  Rectangle(hdc, 1165, 215,1295, 225);  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,1227,360,1305,392);  Ellipse(hdc,1153,360,1229,392);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  //thong  Ellipse(hdc,1160,330,1300,195); //wi tong  Ellipse(hdc,1180,300,1280,210); //wi tong  Rectangle(hdc, 1181, 260,1279, 261);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Rectangle(hdc, 1180, 221,1282, 260);    LineTo(hdc, 1163, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,1105,260,1155,305);  Ellipse(hdc,1305,260,1355,305);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));      //head    Ellipse(hdc,1125,210,1335,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  Ellipse(hdc,1125,209,1335,30); // big h  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,1145,210,1315,70); //small h  Ellipse(hdc,1192,100,1229,60); //ta  Ellipse(hdc,1231,100,1266,60); //ta  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Ellipse(hdc,1219,118,1239,98); //jamug  SelectObject(hdc,CreateSolidBrush(RGB(255,0,0)));  Rectangle(hdc, 1170, 195,1289, 205); //pogkor  Ellipse(hdc,1167,195,1174,205); //jamug  Ellipse(hdc,1285,195,1292,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,0,0)));  Ellipse(hdc,1168,195,1175,205); //jamug  Ellipse(hdc,1284,195,1291,205); //jamug  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Rectangle(hdc, 1198, 205,1262, 260);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SelectObject(hdc,CreateSolidBrush(RGB(255,255,0)));  SelectObject(hdc,CreatePen(PS\_SOLID,2,RGB(0,0,0)));  Ellipse(hdc,1215,190,1245,220);//kading  Ellipse(hdc,1213,197,1247,205);  Rectangle(hdc, 1230, 205,1231,220);  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,1225,206,1235,214);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  SelectObject(hdc,CreateSolidBrush(RGB(0,0,0)));  Ellipse(hdc,1207,85,1220,70); // ta b r  Ellipse(hdc,1238,85,1250,70); // ta b R  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse(hdc,1170,145,1290,170); // m  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  Ellipse(hdc,1170,130,1290,168); //w pagbon    SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  Rectangle(hdc, 1229, 118,1230, 170); //s jamug      //nuad  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1218, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 1215, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1155, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 1217, 141);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1160, 147);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 1241, 120);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1300, 110);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 1240, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1303, 130);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(255,255,255)));  LineTo(hdc, 1240, 140);  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,0,0)));  LineTo(hdc, 1297, 143);      SelectObject(hdc,CreateSolidBrush(RGB(66,143,232)));  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(66,143,232)));  Rectangle(hdc, 1065,130,1082,140);  Rectangle(hdc, 815,130,832,140);  Rectangle(hdc, 565,130,582,140);  Rectangle(hdc, 315,130,332,140);  EndPaint(hwnd,&ps);  return 0;  case WM\_DESTROY:  PostQuitMessage(0);  return 0;  }  return DefWindowProc (hwnd, message, wParam, lParam); |
| ผลการทดลอง |
| การเพิ่มจำนวนโดเรม่อนในที่นี้คือ Coppy Code โดเรม่อนที่เราสร้างเพิ่มอีก 4 ชุด โดย ชุดที่ 1 ในตำแหน่งพิกัดที่ 1 และ 3 จะทำการบวกเพิ่มจากตำแหน่งเดิม + 250  ชุดที่ 2 ในตำแหน่งพิกัดที่ 1 และ 3 จะทำการบวกเพิ่มจากตำแหน่งเดิม + 500  ชุดที่ 3 ในตำแหน่งพิกัดที่ 1 และ 3 จะทำการบวกเพิ่มจากตำแหน่งเดิม + 750  ชุดที่ 4 ในตำแหน่งพิกัดที่ 1 และ 3 จะทำการบวกเพิ่มจากตำแหน่งเดิม + 1000 จะทำให้ด็โดเรม่อนเพิ่มมาอีก สี่ตัว รวมตัวที่เราสร้างเป็น 5 |
| รูป |
|  |