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

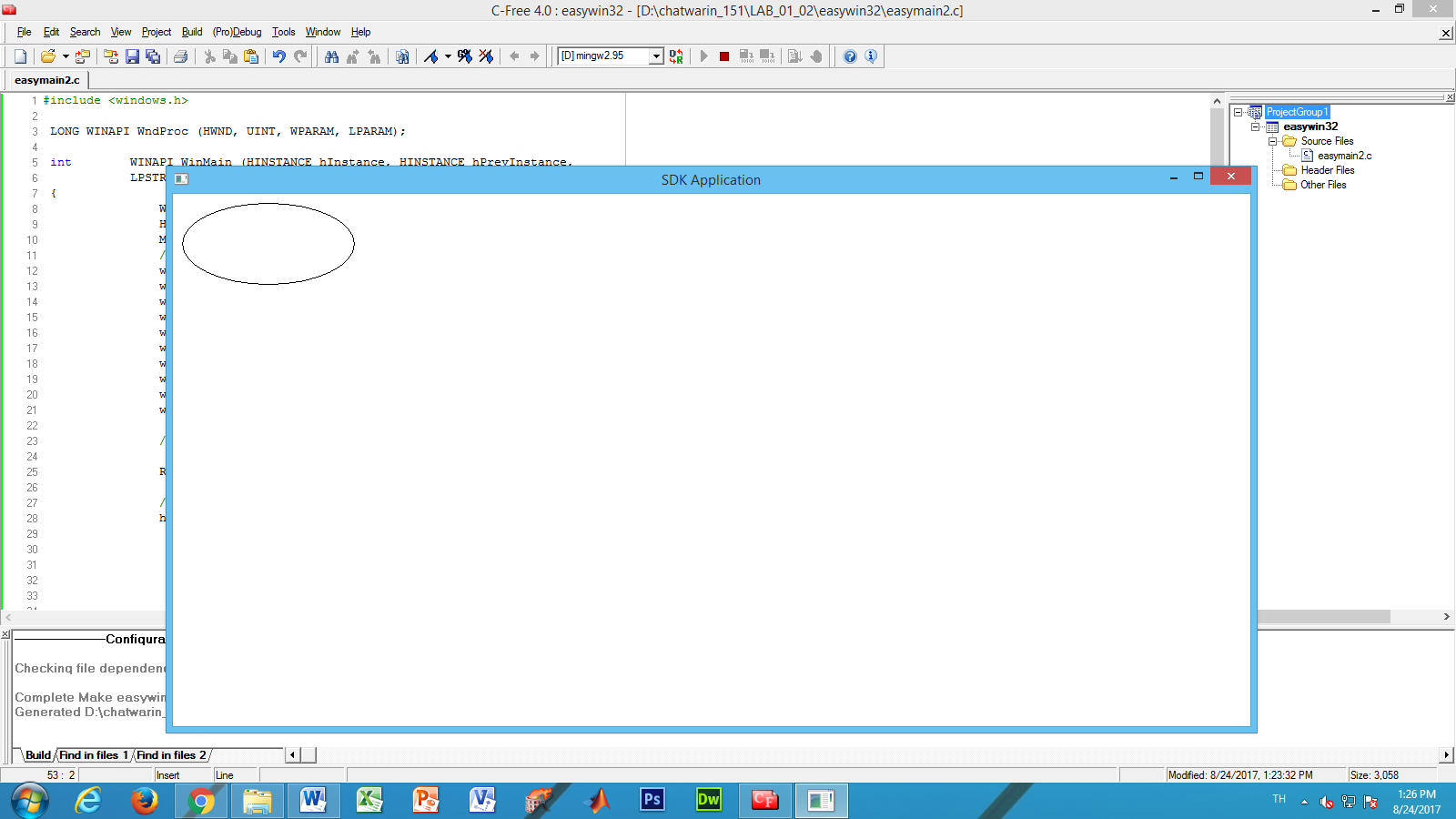
10. เปลี่ยนฟังก์ชั่น Ellipse() เป็นฟังก์ชันวาดรูป

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

11. แก้ไขฟังก์ชั่นในการวาดรูป

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

12. เขียนโปรแกรมวาดรูป Doremon



# บันทึกผล

ตารางการเปลี่ยนค่า WNDCLASS

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); // Background color | wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 5); // Background color |
| ผลการทดลอง | |
| ให้สีของหน้าต่างเป็นไปตามที่ Windows กำหนด จาก color\_window +1 คือสีขาว ปรับเปลี่ยนเป็น +5 จะได้เป็นสีดำ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
|  |  |
| ผลการทดลอง | |
|  | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hCursor = LoadCursor (NULL, IDC\_ARROW); // Cursor handle | wc.hCursor = LoadCursor (NULL,IDC\_NO); // Cursor handle |
| ผลการทดลอง | |
| จากคำสั่งเดิมจะได้เคอร์เซอร์เป็นรูปลูกศร แต่หลังจากเปลี่ยนคำสั่งใหม่เป็น IDC\_NO ผลลัพธ์ที่ได้เคอร์เซอร์เปลี่ยนเป็นสัญลักษณ์ห้าม | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hIcon = LoadIcon (NULL, IDI\_WINLOGO); // Icon handle | wc.hIcon = LoadIcon (0, IDI\_ERROR); // Icon handle |
| ผลการทดลอง | |
| จาก icon ในค่าเดิมคือรูปแอพพลิเคชั่น เราปรับให้รูป icon เป็น error ด้วยคำสั่ง IDI\_ERROR | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.style = 0; // Class style | wc.style = CS\_NOCLOSE; // Class style |
| ผลการทดลอง | |
| จากค่าเดิมที่ปุ่มกดปิดสามารถกดได้ พอเปลี่ยนคำสั่งใหม่เป็น CS\_NOCLOSE แล้วจะไม่สามารถกดปิดได้ | |
| รูป | |
|  | |

ตารางการเปลี่ยนค่าพารามิเตอร์ของฟังก์ชั่น CreateWindow()

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| WS\_OVERLAPPEDWINDOW, // Window style | WS\_EX\_CLIENTEDGE, // Window style |
| ผลการทดลอง | |
| จากคำสั่งเดิมกรอบหน้าต่าง window ปกติมีปุ่มกดย่อลง,ขยาย,ปิดที่มุมบนขวาและมีสัญลักษณ์ไอค่อนที่มุมบนซ้าย แต่หลังจากเปลี่ยนคำสั่งเป็น WS\_EX\_CLIENTEDGE กรอบหน้าต่างก็ไม่มีปุ่มกดต่างๆที่มุมบนขวาและไม่มีสัญลักษณ์ที่มุมบนซ้าย | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| CW\_USEDEFAULT, // Initial width | 640, // Initial width |
| ผลการทดลอง | |
| จากคำสั่งเดิมไม่มีการกำหนดขนาดความกว้างของหน้าต่าง หลังจากเรากำหนดขนาดของความกว้างหน้าต่าง หน้าต่างที่ได้มาจึงมีความกว้างเท่ากับขนาดที่เรากำหนด | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| CW\_USEDEFAULT, // Initial height | 500, // Initial height |
| ผลการทดลอง | |
| จากคำสั่งเดิมไม่มีการกำหนดขนาดความสูงของหน้าต่าง หลังจากเรากำหนดขนาดของความสูงหน้าต่าง หน้าต่างที่ได้มาจึงมีความสูงเท่ากับขนาดที่เรากำหนด | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| "SDK Application", // Window title | "chatwarin", // Window title |
| ผลการทดลอง | |
| จากคำสั่งเดิมจะได้หัวข้อบนหน้าต่างว่า “SDK Application” หลังจากเปลี่ยนข้อความในเครื่องหมาย “ “ เป็น chatwarin แล้ว จะได้หัวข้อบนหน้าต่างตามข้อความที่เราได้พิมพ์ไป | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| CW\_USEDEFAULT, // Horizontal position | 500, // Horizontal position |
| ผลการทดลอง | |
| จากคำสั่งเดิมที่ไม่มีการกำหนดค่าตำแหน่งของหน้าต่างที่จะเด้งขึ้นมาหลังการรันโปรแกรม เมื่อเราเปลี่ยนแปลงค่าให้ ในที่นี้กำหนดค่าที่ 500 ทำให้ตำแหน่งหน้าต่างที่เด้งขึ้นมาอยู่ในตำแหน่งแนวนอนที่ 500 | |
| รูป | |
|  | |

ตารางการเปลี่ยนฟังก์ชั่นในการวาดรูป

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
|  | SetPixel(hdc, 50,50,RGB(250,0,0)); |
| ผลการทดลอง | |
| สร้างจุดได้จากฟังก์ชั่น SetPixel โดยที่ผลแสดงออกจะออกมาในรูปของจุดขนาดเล็ก มีสีแดงตามค่าของสี RGB ที่กำหนด | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 100);  EndPaint (hwnd, &ps);  return 0; | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 200);  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| เป็นการสร้างวงกลมโดยใช้ฟังก์ชั่น Ellipse โดยเริ่มจากการกำหนด hdc, แกน x1, แกน y1, แกน x2, แกน y2 ซึ่งคือตำแหน่งที่ต้องการให้วงกลมแสดงออกบนหน้าจอ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 100);  EndPaint (hwnd, &ps);  return 0; | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 200);  SetPixel(hdc, 50,50,RGB(250,0,0));  Arc(hdc,200,200,300,300,200,200,400,400);  Rectangle(hdc,550,550,300,300);  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| ต้องการสร้างรูปสี่เหลี่ยม โดยใช้ฟังก์ชั่น Rectangle โดยในวงเล็บจะเป็นการกำหนดตำแหน่ง x, y ที่ต้องการให้รูปสี่เหลี่ยมแสดงออกทางหน้าจอ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 100);  EndPaint (hwnd, &ps);  return 0; | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 200);  SetPixel(hdc, 50,50,RGB(250,0,0));  Arc(hdc,200,200,300,300,200,200,400,400);  LineTo(hdc,250,350);  LineTo(hdc,250,10);  LineTo(hdc,150,10);  LineTo(hdc,0,10);  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| ต้องการสร้างรูปสามเหลี่ยม โดยใช้ฟังก์ชั่น LineTo โดยในวงเล็บจะเป็นการกำหนดตำแหน่งที่ต้องการให้รูปสามเหลี่ยมแสดงออกทางหน้าจอ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 100);  EndPaint (hwnd, &ps);  return 0; | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  LineTo(hdc,200,200);  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| ต้องการสร้างเส้นตรง โดยใช้ฟังก์ชั่น LineTo ฟังก์ชั่นเดียวกันสร้างสามเหลี่ยม โดยในวงเล็บจะเป็นการกำหนดตำแหน่ง x, y ที่ต้องการให้เส้นตรงแสดงออกทางหน้าจอ | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 100);  EndPaint (hwnd, &ps);  return 0; | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  Ellipse (hdc, 10, 10, 200, 200);  SetPixel(hdc, 50,50,RGB(250,0,0));  **Arc(hdc,200,200,300,300,200,200,400,400);**  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| ต้องการสร้างเส้นโค้ง โดยใช้ฟังก์ชั่น Arc โดยในวงเล็บจะเป็นการกำหนดตำแหน่งที่ต้องการให้เส้นโค้งแสดงออกทางหน้าจอ ซึ่งฟังก์ชั่นสามารถสร้างวงกลมได้เช่นกัน | |
| รูป | |
|  | |

ตารางการเปลี่ยนค่าพารามิเตอร์ของฟังก์ชั่นในการวาดรูป

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
|  | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  **SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));**  Ellipse (hdc, 10, 10, 200, 200);  SetPixel(hdc, 50,50,RGB(250,0,0));  Arc(hdc,200,200,300,300,200,200,400,400);  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| เป็นฟังก์ชั่นการเปลี่ยนสีของวัตถุที่เราต้องการ โดยวางคำสั่งไว้หน้าวัตถุที่เราต้องการเปลี่ยนสีได้เลย ในทีนี้ต้องการเปลี่ยนววงกลมให้เป็นสีแดงจึงวางไว้หน้าประโยคสร้างวงกลม | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
|  | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));  Ellipse (hdc, 10, 10, 200, 200);  SetPixel(hdc, 50,50,RGB(250,0,0));  **SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,50,150)));**  Arc(hdc,200,200,300,300,200,200,400,400);  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| เป็นฟังก์ชั่นการเปลี่ยนสีขอบเส้นของวัตถุที่เราต้องการ โดยวางคำสั่งไว้หน้าวัตถุที่เราต้องการเปลี่ยนสีได้เลย ในทีนี้ต้องการเปลี่ยนเส้นโค้งให้เป็นสีน้ำเงินจึงวางไว้หน้าประโยคสร้างเส้นโค้ง | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
|  | case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  SetPixel(hdc, 50,50,RGB(250,0,0));  SelectObject(hdc,CreatePen(PS\_SOLID,1,RGB(0,50,150)));  Arc(hdc,200,200,300,300,200,200,400,400);  **SelectObject(hdc,CreatePen(PS\_DASHDOTDOT,2,RGB(0,0,255)));**  LineTo(hdc,200,200);  EndPaint (hwnd, &ps);  return 0; |
| ผลการทดลอง | |
| เป็นฟังก์ชั่นการเปลี่ยนลักษณะของเส้นของวัตถุที่เราต้องการ โดยวางคำสั่งไว้หน้าวัตถุที่เราต้องการเปลี่ยนลักษณะของเส้นได้เลย ในทีนี้ต้องการเปลี่ยนเส้นตรงให้เป็นเส้นทึบจึงวางไว้หน้าประโยคสร้างเส้นตรง | |
| รูป | |
|  | |

|  |  |
| --- | --- |
| ค่าเดิม | ค่าใหม่ |
| wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); // Background color | wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 9); // Background color |
| ผลการทดลอง | |
| ต้องการเปลี่ยนพื้นหลังให้เป็นสีอื่น จึงใช้คำสั่ง wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + ตัวเลขของสี ได้เลย โดยในทีนี้ต้องการพื้นหลังสีฟ้า | |
| รูป | |
|  | |

โปรแกรมวาดรูป Doremon

|  |
| --- |
| โปรแกรม |
| case WM\_PAINT:  hdc = BeginPaint (hwnd, &ps);  SelectObject(hdc,CreateSolidBrush(RGB(0,0,255)));  Ellipse (hdc,50,200,250,400);  Ellipse (hdc,100,100,200,200);  Ellipse (hdc,5,200,57,300);  Ellipse (hdc,244,200,300,305);  SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));  Ellipse (hdc,115,125,185,200);  Ellipse (hdc,80,200,220,350);  Ellipse (hdc,130,140,150,160);  Ellipse (hdc,155,140,175,160);  Ellipse (hdc,5,175,47,205);  Ellipse (hdc,245,175,290,205);  Ellipse (hdc,100,395,150,450);  Ellipse (hdc,150,395,200,450);  Ellipse (hdc,155,145,165,155);  Ellipse (hdc,140,185,165,180);  Ellipse (hdc,100,270,200,320);  SelectObject(hdc,CreateSolidBrush(RGB(50,50,50)));  Ellipse (hdc,140,145,150,155);  Ellipse (hdc,155,145,165,155);  Ellipse (hdc,150,165,155,168);  SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));  Ellipse (hdc,140,200,160,220); |
| ผลการทดลอง |
| จากการทพลองโปรแกรมเราจะสร้างโดเรม่อนจากการวาดวงกลมและวงรีในโปรแกรม ในส่วนของ selectobject แรก คือส่วนที่ใส่สีน้ำเงินเป็นสีพื้นหลัง โดยจะมีส่วนของวงกลมวงตัวและหัวด้านนอกและแขนซ้าย-ขวา selectobject ที่สองคือต้องการให้สีพื้นหลังมีสีขาว โดยจะเป็นส่วนของหัวและตัวด้านใน มือซ้าย-ขวา เท้าซ้าย-ขวา และกรอบตาซ้าย-ขวา selectobject ที่สามคือส่วนที่ต้องการให้สีพื้นหลังของรูปร่างวัตถุมีสีดำ จะประกอบไปด้วยส่วนของตำดำซ้าย-ขวา และจมูก ส่วน selectobject สุดท้าย จะเป็นส่วนที่ต้องการให้พื้นหลังมีสีแดง ซึ่งก็คือกระดอ่งตรงคอของโดเรม่อนนั่นเอง |
| รูป |
|  |

# สรุปผลการทดลอง

# คำถามท้ายการทดลอง

1. อธิบายการทำงานของโปรแกรมในข้อ 4

Ans. อันดับแรกเริ่มจากส่วนที่ 1 การกำหนด window ที่เราต้องการ โดยกำหดนได้ตามความต้องการของเรา ไม่ว่าจะเป็นสีพื้นหลัง โดยคำสั่ง wc.hbrBackground = (HBRUSH) (COLOR\_WINDOW + 1); หรือกำหนดรูปร่างลักษณะของเคอร์เซอร์ก็ได้ด้วยคำสั่ง wc.hCursor = LoadCursor (NULL, IDC\_ARROW); ส่วนที่ 2 จะเป็นการลงชื่อเข้าใช้งานคลาสที่ตั้งขึ้นมา ในส่วนที่ 3 จะเป็นการสร้าง window ขึ้นมา ซึ่งสามารถตั้งชื่อของ window ได้ด้วยคำสั่ง hwnd = CreateWindow ( ตามด้วยชื่อของ window ที่เราต้องการตั้งใส่ในเครื่องหมาย “ ” และยังสามารถกำหนดรูปแบบของกรอบหน้าต่าง window ได้ด้วยคำสั่ง WS\_OVERLAPPEDWINDOW สามารถเปลี่ยนรูปแบบได้ตามที่ต้องการ มีการกำหนดขนาดความกว้างและความยาวของหน้าต่าง ทั้งยังกำหนดตำแหน่งของ window หลังการรันโปรแกรมได้อีกด้วย ในส่วนที่ 4 จะเกี่ยวกับการอัพเดทข้อมูล window โดยมีคำสั่งให้โชว์การเปลี่ยนแปลงของ window ในส่วนที่ 5 จะเป็นส่วนของการแปลคีย์ของแมสเสจและเป็นส่วนทีมีการเขียนโปรแกรมเพื่อวาดวัตถุที่เราต้องการได้หลังจากคำสั่ง case WM\_PAINT: ได้

1. บอกพารามิเตอร์ของ WNDCLASS

Ans. HWND hwnd;

MSG msg; กำหนดตัวแปรเก็บค่าที่ได้รับเข้า

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

1. บอกพารามิเตอร์ และหน้าที่ของฟังก์ชั่น CreateWindow

Ans. hwnd = CreateWindow (

"MyWndClass", // WNDCLASS name เป็นชื่อแถบต้นแบบ

" SDK Application", // Window title ชื่อแถบที่ต้องการตั้งขึ้น

WS\_OVERLAPPEDWINDOW, // Window style กำหนดชนิดขอบหน้าต่าง

CW\_USEDEFAULT, // Horizontal position กำหนดตำแหน่งในแนวนอน

CW\_USEDEFAULT, // Vertical position กำหนดตำแหน่งในแนวตั้ง

CW\_USEDEFAULT, // Initial width กำหนดค่าความกว้าง

CW\_USEDEFAULT, // Initial height กำหนดค่าความสูง

HWND\_DESKTOP, // Handle of parent window

NULL, // Menu handle

hInstance, // Application's instance handle

NULL // Window-creation data

);

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

Ans.

RegisterClass() คือ การลงทะเบียนคลาสหน้าต่างสำหรับการใช้งานหลังจากการเรียกใช้ CreateWindow() คือ การสร้างหน้าต่างเพื่อใช้งาน

ShowWindow() คือ การตั้งค่าสถานการณ์แสดงของหน้าต่างที่ต้องการ

UpdateWindow() คือ ฟังก์ชั่นที่ปรับปรุงหน้าต่าง

TranslateMessage() คือ แปลข้อความเสมือนคีย์เป็นข้อความตัวอักษร

DispatchMessage() คือ ส่งข้อความไปยังขั้นตอนของหน้าต่าง โดยทั่วไปจะใช้เพื่อส่งข้อความที่ดึงข้อมูล

1. บอกชื่อและหน้าที่ WM\_MESSAGE มา 10 ชื่อ

Ans. WM\_QUIT

WM\_LBUTTONDOWN

WM\_PAINT

WM\_CLOSE

WM\_SIZE

WM\_MOVE

WM\_CUT

WM\_COPY

WM\_PASTE

WM\_UNDO

1. เขียนโปรแกรมสร้าง Doraemon จำนวน 5 ตัว

case WM\_PAINT:

hdc = BeginPaint (hwnd, &ps);

ตัวที่ 1

SelectObject(hdc,CreateSolidBrush(RGB(0,0,255)));

Ellipse (hdc,50,200,250,400);

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

Ellipse (hdc,5,200,57,300);

Ellipse (hdc,244,200,300,305);

SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));

Ellipse (hdc,115,125,185,200);

Ellipse (hdc,80,200,220,350);

Ellipse (hdc,130,140,150,160);

Ellipse (hdc,155,140,175,160);

Ellipse (hdc,5,175,47,205);

Ellipse (hdc,245,175,290,205);

Ellipse (hdc,100,395,150,450);

Ellipse (hdc,150,395,200,450);

Ellipse (hdc,155,145,165,155);

Ellipse (hdc,140,185,165,180);

Ellipse (hdc,100,270,200,320);

SelectObject(hdc,CreateSolidBrush(RGB(50,50,50)));

Ellipse (hdc,140,145,150,155);

Ellipse (hdc,155,145,165,155);

Ellipse (hdc,150,165,155,168);

SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));

Ellipse (hdc,140,200,160,220);

ตัวที่ 2

SelectObject(hdc,CreateSolidBrush(RGB(0,0,255)));

Ellipse (hdc,400,100,500,200);

Ellipse (hdc,350,200,550,400);

Ellipse (hdc,305,200,357,300);

Ellipse (hdc,544,200,600,305);

SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));

Ellipse (hdc,305,175,347,205);

Ellipse (hdc,450,395,500,450);

Ellipse (hdc,400,395,450,450);

Ellipse (hdc,545,175,590,205);

Ellipse (hdc,415,125,485,200);

Ellipse (hdc,380,200,520,350);

Ellipse (hdc,430,140,450,160);

Ellipse (hdc,455,140,475,160);

Ellipse (hdc,440,185,465,180);

Ellipse (hdc,400,270,500,320);

SelectObject(hdc,CreateSolidBrush(RGB(50,50,50)));

Ellipse (hdc,440,145,450,155);

Ellipse (hdc,455,145,465,155);

Ellipse (hdc,450,165,455,168);

SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));

Ellipse (hdc,440,200,460,220);

ตัวที่ 3

SelectObject(hdc,CreateSolidBrush(RGB(0,0,255)));

Ellipse (hdc,700,100,800,200);

Ellipse (hdc,650,200,850,400);

Ellipse (hdc,605,200,657,300);

Ellipse (hdc,844,200,900,305);

SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));

Ellipse (hdc,605,175,647,205);

Ellipse (hdc,750,395,800,450);

Ellipse (hdc,700,395,750,450);

Ellipse (hdc,845,175,890,205);

Ellipse (hdc,715,125,785,200);

Ellipse (hdc,680,200,820,350);

Ellipse (hdc,730,140,750,160);

Ellipse (hdc,755,140,775,160);

Ellipse (hdc,740,185,765,180);

Ellipse (hdc,700,270,800,320);

SelectObject(hdc,CreateSolidBrush(RGB(50,50,50)));

Ellipse (hdc,740,145,750,155);

Ellipse (hdc,755,145,765,155);

Ellipse (hdc,750,165,755,168);

SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));

Ellipse (hdc,740,200,760,220);

ตัวที่ 4

SelectObject(hdc,CreateSolidBrush(RGB(0,0,255)));

Ellipse (hdc,1000,100,1100,200);

Ellipse (hdc,950,200,1150,400);

Ellipse (hdc,905,200,957,300);

Ellipse (hdc,1144,200,1200,305);

SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));

Ellipse (hdc,905,175,947,205);

Ellipse (hdc,1050,395,1100,450);

Ellipse (hdc,1000,395,1050,450);

Ellipse (hdc,1145,175,1190,205);

Ellipse (hdc,1015,125,1085,200);

Ellipse (hdc,980,200,1120,350);

Ellipse (hdc,1030,140,1050,160);

Ellipse (hdc,1055,140,1075,160);

Ellipse (hdc,1040,185,1065,180);

Ellipse (hdc,1000,270,1100,320);

SelectObject(hdc,CreateSolidBrush(RGB(50,50,50)));

Ellipse (hdc,1040,145,1050,155);

Ellipse (hdc,1055,145,1065,155);

Ellipse (hdc,1050,165,1055,168);

SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));

Ellipse (hdc,1040,200,1060,220);

ตัวที่ 5

SelectObject(hdc,CreateSolidBrush(RGB(0,0,255)));

Ellipse (hdc,300,400,400,500);

Ellipse (hdc,250,500,450,700);

Ellipse (hdc,205,500,257,600);

Ellipse (hdc,444,500,500,605);

SelectObject(hdc,CreateSolidBrush(RGB(255,255,255)));

Ellipse (hdc,315,425,385,500);

Ellipse (hdc,280,500,420,650);

Ellipse (hdc,330,440,350,460);

Ellipse (hdc,355,440,375,460);

Ellipse (hdc,205,475,247,505);

Ellipse (hdc,445,475,490,505);

Ellipse (hdc,300,695,350,750);

Ellipse (hdc,350,695,400,750);

Ellipse (hdc,340,485,365,480);

Ellipse (hdc,300,570,400,620);

SelectObject(hdc,CreateSolidBrush(RGB(50,50,50)));

Ellipse (hdc,340,445,350,455);

Ellipse (hdc,355,445,365,455);

Ellipse (hdc,350,465,355,468);

SelectObject(hdc,CreateSolidBrush(RGB(250,0,0)));

Ellipse (hdc,340,500,360,520);

EndPaint (hwnd, &ps);

return 0;

