

## รายการอ้างอิง

1. Lee Adams. SuperCharged C Graphics A Programmer's Source Code Toolbox. : Windcrest Books, 1990.
2. Ben Ezzell. Graphics Programming In Turbo C 2.0. : Addison-Wesley Publishing Company, 1989.
3. Herbert Schidlt. Turbo C/C++ the complete reference. : Osborne Mcgraw-Hill, 1990.
4. Ben Ezzell. Turbo C++ Programming. : Addison-Wesley Publishing Company, 1990.
5. Donald Hearn, M. Pauline Baker. Computer Graphics. : Prentice Hall, 1986.
6. Julio Sanchez. Graphics Design And Animation On The IBM Micro Computers. : Prentice Hall, 1990.
7. Lee Adams. High-Performance Graphics in C Animation and Simulation. : Windcrest Books, 1988.
8. Microsoft Corporation. Microsoft Windows Software Development Kit Programming Tools Version 2.0. : Microsoft Corporation.
9. Rebecca Wirfs-Brock, Brian Wilkerson, Lauren Wiener. Designing object-oriented software. : Prentice Hall , 1990.
10. Robert Lafore. The Waite Group's C Programming Using Turbo C++. : The Waite's Group, 1990.
11. Scott Robert Ladd. C++ Techniques and applications. : Prentice Hall, 1990.
12. Steve Rimmer. Bit-Mapped Graphics. : Wincrest Books, 1990.
13. Steven Harrington. Computer Graphics A Programming Approach. : McGRAW-HILL 2nd ed., 1987.
14. Taeuber, Clemens. Hijak Version 2.0. : Inset systems, 1991.
15. กองบรรณาธิการ. OOP ความหวังของการเก็บของเก่ามาใช้ใหม่. ไมโครคอมพิวเตอร์ (ตุลาคม 2532): 238-245.
16. จิระ จริงจิตร. มาจัดเก็บภาพแบบ TIFF กันดีกว่า. คอมพิวเตอร์วิวิ ปีที่ 8 (เมษายน 2534): 157-167.
17. บุญเลิศ เอี่ยมทัศน. แนะนำเทอร์โบ C++. ไมโครคอมพิวเตอร์ (ตุลาคม 2533): 315-323.
18. ประกาศิต ชาติบุรุษ. OOP. คอมพิวเตอร์วิวิ ปีที่ 7 (มกราคม 2533): 99-104.
19. ดร.ไพศาล สงวนหมุ่ม. กราฟิกบนไมโครคอมพิวเตอร์. ไมโครคอมพิวเตอร์ (มกราคม-กุมภาพันธ์ 2533): 269-283.

20. ยืน ภูววรรณ. กราฟิกเทคโนโลยี. ไมโครคอมพิวเตอร์ (ตุลาคม 2531): 182-195.
21. ศรายุทธ ฉายสุริยะ. การพัฒนาโปรแกรมสำหรับจัดทำเอกสารที่มีภาพประกอบ. วิทยานิพนธ์ปริญญา  
มหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2534.
22. อาจารย์ ลัตยารักษ์. เทคนิคการเขียนกราฟิกบนเทอร์มินัล. บริษัทซีเอ็ดยูเคชั่น.
23. \_\_\_\_\_. ภาษาไทยกราฟิกทึส สมอ. ไมโครคอมพิวเตอร์ (พฤศจิกายน 2535) 299-306.
24. นพ.อภิชัย มงคล. โปรแกรมแปลงไฟล์กราฟิก (โปรแกรม The Graphics Link Plus และโปรแกรม  
HiJaak). คอมพิวเตอร์วิว ปีที่ 8 (พฤศจิกายน 2533): 181-188.



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก.

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



## คู่มือสำหรับผู้พัฒนาโปรแกรม

เครื่องมือซอฟต์แวร์ที่พัฒนาขึ้นมาได้จัดทำไว้ในรูปแบบของคลังโปรแกรม โดยจัดเก็บไว้ในแฟ้มข้อมูลชื่อ cupaint.lib เนื่องจากการพัฒนาโปรแกรมวาดภาพนี้ ใช้ตัวแปลโปรแกรม TURBO C++ ดังนั้นผู้ที่สนใจควรมีความรู้ทางด้านการพัฒนาโปรแกรมทางด้านกราฟิกด้วยตัวแปลโปรแกรม TURBO C++ สำหรับผู้ที่สนใจ และยังไม่มีความรู้ทางด้านการเขียนโปรแกรมกราฟิก ด้วยตัวแปลโปรแกรม TURBO C++ สามารถศึกษาได้จาก [2] และ [3]

สำหรับโครงสร้างข้อมูล และแมโคร (Macro) ต่าง ๆ ที่กำหนดไว้ในแฟ้มข้อมูล "graphics.h" ของตัวแปลโปรแกรม TURBO C++ ผู้วิจัยจะขอนำมาใช้โดยไม่กล่าวในรายละเอียด ซึ่งผู้ที่สนใจสามารถที่จะศึกษาได้จากเอกสารอ้างอิงที่กล่าวมาแล้ว

### 1. คลาสเกี่ยวกับเมาส์

หน้าที่ : ควบคุมการทำงานของเมาส์

ชื่อคลาส : mouse\_class

แฟ้มข้อมูลที่กำหนดนิยามของคลาส : mouse.hpp

โครงสร้างข้อมูลที่กำหนดในคลาสนี้:

```
ms_result ซึ่งมีโครงสร้างข้อมูลดังต่อไปนี้
struct ms_result
{
    int presents;
    int buttons;
};
ตัวแปร presents จะแสดงว่าในระบบมีโปรแกรมขับเมาส์อยู่หรือไม่ ถ้าไม่มี ตัวแปร presents
จะมีค่าเป็น 0
ตัวแปร buttons จะแสดงถึงจำนวนปุ่มของเมาส์
```

ms\_status ซึ่งมีโครงสร้างข้อมูลดังต่อไปนี้

```
struct ms_status
{
    int button_count,
    xaxis, yaxis;
};
```

ตัวแปร button\_count จะนับจำนวนครั้งที่มีการกดปุ่มเมาส์

ตัวแปร xaxis และ yaxis จะรายงานตำแหน่งของตัวชี้ของเมาส์

วิธีการต่าง ๆ :

```
ms_result ms_reset( void )
```

ใช้สำหรับกำหนดค่าเริ่มต้นให้แก่เมาส์ และตรวจสอบว่าในระบบมี โปรแกรมขับเมาส์อยู่หรือไม่ ตัวอย่างการใช้งาน :

```
mouse_class mouse_obj;
ms_result result;
result = mouse_obj.ms_reset();
if( !result.present )
{
    printf( "Error mouse driver not installed" );
    exit( 1 );
}
```

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

```
ms_status ms_pos( void )
```

ใช้สำหรับตรวจสอบตำแหน่งของตัวชี้ของเมาส์ และสถานะของเมาส์ เช่นมีการกดปุ่มเมาส์หรือไม่ ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
ms_status position;
position = mouse_obj.ms_pos();
if( position.button_count )
{
    if( position.xaxis == 400 && position.yaxis ==
        300 )
    {
        ...
    }
}
```

```
ms_status ms_pressed( int button )
```

ใช้สำหรับตรวจสอบว่ามีการกดปุ่มเมาส์ ที่ต้องการตรวจสอบหรือไม่ สำหรับอาร์กิวเมนต์ที่จะต้องส่งให้กับวิธีการนี้คือ ปุ่มที่ต้องการจะตรวจสอบ โดยได้กำหนดแมโครสำหรับปุ่มต่าง ๆ ไว้ในแฟ้มข้อมูล "mouse.hpp" ดังนี้

- 1) BUTTON\_LEFT หมายถึงปุ่มซ้ายของเมาส์
- 2) BUTTON\_RIGHT หมายถึงปุ่มขวาของเมาส์
- 3) BUTTON\_MIDDLE หมายถึงปุ่มกลางของเมาส์

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
ms_status status;
status = mouse_obj.ms_pressed( BUTTON_LEFT );
if( status.button_count )
{
    ...
}
```

```
ms_status ms_released( int button )
```

ใช้สำหรับตรวจสอบการปล่อยปุ่มเมาส์ (ดู ms\_pressed() )

```
void ms_show( int showstat )
```

ใช้สำหรับแสดง หรือซ่อนตัวชี้ตำแหน่งของเมาส์ โดยอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ สิ่งที่จะบอกว่าจะซ่อน หรือแสดงตัวชี้ตำแหน่งของเมาส์ โดยได้กำหนดแมโครสำหรับการซ่อน หรือแสดงตัวชี้ตำแหน่งของเมาส์ ไว้ในแฟ้มข้อมูล "mouse.hpp" ดังนี้

- 1) FALSE หมายถึงซ่อนตัวชี้ตำแหน่งของเมาส์
- 2) TRUE หมายถึงแสดงตัวชี้ตำแหน่งของเมาส์

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
mouse_obj.ms_show( FALSE );
```

```
void ms_moveto( int xaxis, int yaxis )
```

ใช้สำหรับเลื่อนตัวชี้ตำแหน่งของเมาส์ ไปยังตำแหน่งที่ต้องการบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ก็คือตำแหน่งของหน้าจอ ที่ต้องการจะให้ตัวชี้ตำแหน่งของเมาส์เลื่อนไป โดยตำแหน่งจะต้องกำหนดเป็นพิกเซล

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
mouse_obj.ms_moveto( 100, 100 );
```

```
void ms_x_limit( int min_x, int max_x )
```

ใช้สำหรับการกำหนดตำแหน่งที่อนุญาตให้เมาส์เคลื่อนที่ไปได้ตามแนวแกน X โดยอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งในแนวแกน X ที่มากที่สุด และน้อยที่สุด ที่จะอนุญาตให้เมาส์เคลื่อนที่ไป โดยตำแหน่งจะต้องกำหนดเป็นพิกเซล

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
mouse_obj.ms_x_limit( 0, 504 );
```

```
void ms_y_limit( int min_y, int max_y )
```

ใช้สำหรับการกำหนดตำแหน่งที่อนุญาตให้เมาส์เคลื่อนที่ไปได้ตามแนวแกน Y (ดูms\_x\_limit() )

```
void ms_speed( int speed )
```

ใช้สำหรับกำหนดความเร็วของเมาส์ โดยอาร์กิวเมนต์ที่จะต้องส่งให้กับวิธีการนี้คือ ความเร็วของเมาส์ที่ต้องการ โดยความเร็วของเมาส์สามารถกำหนดได้ในช่วง 0 ถึง 32,767

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
mouse_obj.ms_speed( 19 );
```

```
void ms_conceal( int left, int top, int right, int bottom )
```

ใช้สำหรับกำหนดบริเวณที่ไม่ต้องการให้มีการแสดงตัวชี้ของเมาส์ โดยอาร์กิวเมนต์ที่จะต้องส่งให้กับวิธีการนี้คือ บริเวณที่ไม่ต้องการให้มีการแสดงตัวชี้ของเมาส์ โดยตำแหน่งต่าง ๆ จะกำหนดเป็นพิกเซล

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
mouse_obj.ms_conceal( 50, 50, 200, 200 );
```

```
int ms_hit( void )
```

ใช้สำหรับตรวจสอบว่ามีการกดปุ่มเมาส์หรือไม่

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
if( mouse_obj.ms_hit() )
{
    ...
}
```

```
int ms_left_pressed( void )
```

ใช้สำหรับตรวจสอบว่ามีการกดปุ่มซ้ายของเมาส์หรือไม่

ตัวอย่างการใช้งาน

```
mouse_class mouse_obj;
if( mouse_obj.ms_left_pressed )
{
    ...
}
```



```
int ms_right_pressed( void )
```

ใช้สำหรับตรวจสอบการกดปุ่มขวาของเมาส์ (ดู ms\_left\_pressed() )

```
int ms_left_released( void )
```

ใช้สำหรับตรวจสอบการปล่อยปุ่มซ้ายของเมาส์ (ดู ms\_left\_pressed() )

```
int ms_right_released( void )
```

ใช้สำหรับตรวจสอบการปล่อยปุ่มขวาของเมาส์ (ดู ms\_left\_pressed() )

## 2. คลาสเกี่ยวกับกราฟิกเมาส์

คลาสนี้เป็นดีไวส์คลาสของคลาส mouse\_class

หน้าที่ : ควบคุมการทำงานของเมาส์ในภาวะกราฟิก

ชื่อคลาส : g\_mouse\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : gmouse.hpp

วิธีการ :

```
void set_cursor( g_cursor this_cursor )
```

ใช้สำหรับการกำหนดรูปแบบต่าง ๆ ของตัวชี้ตำแหน่งของเมาส์ โดยอาร์กิวเมนต์ที่จะต้องส่งให้วิธีการนี้คือ รูปแบบของเคอร์เซอร์ที่ต้องการ รูปแบบของเคอร์เซอร์ได้กำหนดไว้เป็นแมโครในแฟ้มข้อมูล "gmouse.hpp" ดังนี้

- 1) ARROW\_CUR รูปลูกศร
- 2) CHECK\_CUR รูปเครื่องหมายถูก
- 3) CROSS\_CUR รูปครอสแฮร์ (Cross-Hairs)
- 4) GLOVE\_CUR รูปมือ
- 5) IBEAM\_CUR รูปไอบีม (Ibeam)
- 6) HOURGLASS\_CUR รูปนาฬิกา
- 7) MOUSE\_CUR รูปเมาส์
- 8) PENCIL\_CUR รูปดินสอ
- 9) ERASER\_CUR รูปยางลบ

ตัวอย่างการใช้งาน

```
g_mouse_class gmouse;
gmouse.set_cursor( ARROW_CUR );
```

## 3. คลาสเกี่ยวกับแป้นพิมพ์

หน้าที่ : ควบคุมการทำงานของแป้นพิมพ์

ชื่อคลาส : kb\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : keyboard.hpp

วิธีการ :

```
int chk_key( void )
```

ใช้สำหรับตรวจสอบว่ามีการกดแป้นใดบนแป้นพิมพ์

ตัวอย่างการใช้งาน

```
kb_class keyboard;
int key;
key = keyboard.chk_key();
```

```
int alt_map( int key )
```

ตรวจสอบการกดแป้น Alt ร่วมกับแป้นที่ต้องการหรือไม่ โดยอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตัวอักษรของแป้นที่ต้องการตรวจสอบว่ามีการกดร่วมกับแป้น Alt โดยวิธีการนี้จะส่งค่า รหัสแอสกีของแป้นที่กดร่วมกับแป้น Alt กลับไปในกรณีที่มีการกดแป้นนั้นร่วมกับแป้น Alt ในกรณีที่ไม่ได้มีการกดแป้นนั้นร่วมกับแป้น Alt วิธีการนี้จะส่งค่า 0 กลับไป

ตัวอย่างการใช้งาน

```
kb_class keyboard;
int ret_key;
ret_key = keyboard.alt_map( 'A' );
```

## 4. คลาสเกี่ยวกับการพิมพ์รูปภาพ

หน้าที่ : พิมพ์รูปภาพออกทางเครื่องพิมพ์

ชื่อคลาส : printerclass

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : printer.hpp

วิธีการ :

```
virtual void printpicture( char far *p, int width, int lines )
```

เป็นฟังก์ชันเสมือน (virtual function) เพื่อให้ดีโพรซีจอร์ของ คลาสนี้สามารถเรียกใช้วิธีการนี้ได้โดยอาร์กิวเมนต์สำหรับวิธีการนี้คือตำแหน่งของข้อมูลรูปภาพในหน่วยความจำ ความกว้างของรูปภาพ และจำนวนสแกนไลน์ของรูปภาพ

## 5. คลาสสำหรับเครื่องพิมพ์ EPSON FX-80

คลาสนี้เป็นดีไวส์คลาสของคลาส printpicclass

หน้าที่ : ควบคุมเครื่องพิมพ์ EPSON FX-80

ชื่อคลาส : epsonfx80class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : printer.hpp

วิธีการ :

```
void printpicture( char far *p, int width, int lines )
```

ใช้สำหรับการพิมพ์รูปภาพออกทางเครื่องพิมพ์ (ดูข้อ 4)

หมายเหตุ

โดยทั่วไปจะไม่เรียกวิธีการนี้โดยตรง ถ้าต้องการจะพิมพ์รูปภาพ ควรจะเรียกจากฟังก์ชัน print\_picture() ซึ่งมีต้นแบบ (Prototype) ของฟังก์ชันดังต่อไปนี้

```
void print_picture( printerclass *printer_obj, char far *p,int width, int lines );
```

ซึ่งการเรียกใช้ฟังก์ชันนี้ในการพิมพ์รูปภาพ จะมีประโยชน์ในกรณีที่ต่อไป อาจจะมีการเพิ่มเครื่องพิมพ์รุ่นต่าง ๆ เข้าไปในระบบจะทำให้การ เรียกใช้ฟังก์ชันการพิมพ์รูปภาพทำได้โดยสะดวก

ตัวอย่างการใช้งาน

```
epsonfx80class *printer_obj = new epsonfx80class;
```

```
// ดู setprintmode()
```

```
printer_obj->setprintmode(6);
```

```
print_picture( printer_obj, largepic, width, depth );
```

```
void setprintmode( int mode )
```

ใช้สำหรับกำหนดภาวะสำหรับการพิมพ์ภาพกราฟิก โดยอาร์กิวเมนต์ที่ส่งให้กับวิธีการนี้คือ รหัสของภาวะที่จะส่งเครื่องพิมพ์ให้พิมพ์ภาพ

ตัวอย่างการใช้งาน

```
epsonfx80class printer;
```

```
printer.setprintmode( 6 );
```

```
void p_string( char *s )
```

ใช้สำหรับการพิมพ์สายอักขระ (String) ออกทางเครื่องพิมพ์

ตัวอย่างการใช้งาน

```
epsonfx80class printer_obj;
```

```
printer_obj.p_string( "TEST" );
```

```
void p_char( int c )
```

ใช้สำหรับพิมพ์อักขระออกทางเครื่องพิมพ์

ตัวอย่างการใช้งาน

```
epsonfx80class printer_obj;
printer_obj.p_char( 'C' );
```

#### 6. คลาสสำหรับการจัดการจอภาพ

หน้าที่ : ควบคุมการทำงานของจอภาพ

ชื่อคลาส : viewport\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : viewport.hpp

วิธีการ :

```
int open_window( int left, int top, int right, int bottom,
                int bor_clr, int p_clr );
```

ใช้สำหรับเปิดวินโดวโดยเก็บข้อมูลของจอภาพในตำแหน่งที่จะเปิด วินโดวไว้ด้วยอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ได้แก่ จุดมุมบนด้านซ้าย และมุมล่างด้านขวาของตำแหน่งที่จะเปิดวินโดว (เป็นพิกเซล) สีของขอบของวินโดว และสีที่จะแสดงในวินโดว ในกรณีที่หน่วยความจำไม่พอ สำหรับการเก็บภาพในตำแหน่งที่จะเปิดวินโดววิธีการนี้จะส่งค่า -1 กลับมาและไม่เปิดวินโดวให้

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewport_obj.open_window( 50, 50, 100, 100 );
```

```
void open_window1( int left, int top, int right, int bottom, int bor_clr = BLACK, int p_clr =
                  WHITE )
```

ใช้สำหรับเปิดวินโดวโดยไม่จัดเก็บข้อมูลของภาพในตำแหน่งที่จะเปิดวินโดว

(ดู open\_window() )

```
void close_window( int left, int top, int right, int bottom )
```

ใช้สำหรับการปิดวินโดวที่เปิดโดยใช้วิธีการ open\_window() โดยอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบบริเวณที่ต้องการจะปิดวินโดว (ดู open\_window)

ตัวอย่างการใช้งาน

```
view_portclass viewport_obj;
viewport_obj.close_window( 50, 50, 100, 100 );
```

```
void close_window1 ( int left, int top, int right, int bottom )
```

ใช้สำหรับการปิดวินโดว์ที่เปิดโดยใช้วิธีการ open\_window1() (ดู close\_window)

```
void backup_viewport( void )
```

ใช้สำหรับเก็บค่าของวิวพอร์ตเดิม ก่อนที่จะกำหนดค่าวิวพอร์ตใหม่

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
```

```
viewport_obj.backup_viewport();
```

```
void restore_viewport( void )
```

ใช้สำหรับนำค่าวิวพอร์ตที่เก็บไว้โดยวิธีการ backup\_viewport() กลับมาใช้ใหม่

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
```

```
viewport_obj.restore_viewport();
```

```
void backup_linestyle( void )
```

ใช้สำหรับเก็บค่ารูปแบบของเส้น ก่อนที่จะกำหนดค่ารูปแบบของเส้นใหม่

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
```

```
viewport_obj.backup_linestyle();
```

```
void restore_linestyle( void )
```

ใช้สำหรับการนำค่ารูปแบบของเส้นที่เก็บไว้โดยวิธีการ backup\_linestyle() กลับมาใช้ใหม่

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
```

```
viewport_obj.restore_linestyle();
```

```
void backup_fillstyle( void )
```

ใช้สำหรับเก็บค่าลวดลายของการระบายสี ก่อนที่จะกำหนดลวดลายใหม่

(ดู backup\_linestyle() )

```
void restore_fillstyle( void )
```

ใช้สำหรับการนำลวดลายของการระบายสีที่เก็บไว้โดยวิธีการ backup\_fillstyle() กลับมาใช้ใหม่ (ดู restore\_linestyle() )

```
void backup_color( void )
```

ใช้สำหรับเก็บค่าของสี ก่อนที่จะกำหนดค่าของสีใหม่ (ดู backup\_linestyle() )

```
void restore_color( void )
```

ใช้สำหรับการนำค่าของสีที่เก็บไว้โดยวิธีการ backup\_color() กลับมาใช้ใหม่ (ดู restore\_linestyle() )

```
viewporttype get_viewinfo( void )
```

ใช้สำหรับการรายงานค่าของวิวพอร์ตโดยค่าที่ส่งกลับ จะส่งกลับผ่าน ตัวแปรที่มีโครงสร้างข้อมูลแบบ viewporttype ที่กำหนดไว้ในแฟ้มข้อมูล "graphics.h" ของตัวแปลโปรแกรม TURBO C++ ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewporttype viewinfo;
viewinfo = viewport_obj.get_viewinfo();
```

```
int setup_mode ( char driver_set )
```

ใช้สำหรับการเข้าสู่ภาวะกราฟิก โดยอาร์กิวเมนต์ที่ส่งให้กับวิธีการนี้ คือภาวะกราฟิกที่ต้องการใช้ ในกรณีที่ต้องการให้โปรแกรมตรวจสอบเองว่าควรจะเข้าสู่ภาวะกราฟิกแบบใด ให้ส่งอาร์กิวเมนต์เป็น ' ให้กับวิธีการนี้

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewport_obj.setup_mode( ' ' );
```

```
void shutdown_graphics(void)
```

ใช้สำหรับการออกจากภาวะกราฟิก

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewport_obj.shutdown_graphics();
```

```
int backup( int left, int top, int right, int bottom )
```

ใช้สำหรับเก็บข้อมูลของจอภาพเข้าสู่หน่วยความจำ ในกรณีที่หน่วยความจำไม่พอสำหรับเก็บข้อมูลของจอภาพวิธีการนี้จะส่งค่า -1 กลับมาและไม่เก็บข้อมูลของจอภาพอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดมุมบนด้านซ้าย และมุมล่างด้านขวาของบริเวณของจอภาพที่ต้องการจัดเก็บ โดยอาณาบริเวณจะพิจารณาเป็นพิกเซล

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewport_obj.backup( 50, 50, 100, 100 );
```

```
void clear_backup( void )
```

ใช้สำหรับลบข้อมูลรูปภาพที่เก็บจากวิธีการ backup() ออกจากหน่วยความจำ

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewport_obj.clear_backup();
```

```
int check_backup( void )
```

ใช้สำหรับตรวจสอบว่ามีการเก็บข้อมูลภาพเข้าสู่หน่วยความจำหรือไม่

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
if( viewport_obj.check_backup() )
{
    ...
}
```

```
void restore( int left, int top )
```

ใช้สำหรับเขียนข้อมูลของจอภาพจากหน่วยความจำ ลงจอภาพในตำแหน่งที่ต้องการอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของจอภาพที่ต้องการจะเขียนข้อมูลลงไป โดยตำแหน่งของจอภาพจะระบุเป็นพิกเซล

ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewport_obj.restore( 50, 50 );
```

```
void set_color( int color )
```

ใช้สำหรับเลือกสีที่ต้องการใช้ โดยอาร์กิวเมนต์ที่จะต้องส่งให้กับวิธีการนี้คือค่าของสีที่ต้องการ ตัวอย่างการใช้งาน

```
viewport_class viewport_obj;
viewport_obj.set_color( BLACK );
```

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



### 7. คลาสสำหรับการจัดการการรับและแสดงข้อความ

คลาสนี้เป็นดีโพรไฟคลาสของคลาส viewport\_class

หน้าที่ : รับ และแสดงข้อความบนจอภาพ

ชื่อคลาส : text\_class

แฟ้มข้อมูลที่กำหนดนิยามของคลาส : text.hpp

วิธีการ :

```
unsigned char *read_str( int column, int row, int count,
                        unsigned char tt_mode, int b_color )
```

ใช้สำหรับอ่านข้อความทั้งภาษาไทย และภาษาอังกฤษจากแป้นพิมพ์ อาร์กิวเมนต์ที่ต้องส่ง ให้  
กับวิธีการนี้คือ ตำแหน่งที่ต้องการจะรับข้อมูล (พิจารณาเป็นคอลัมน์ และแถว โดยตำแหน่งของคอลัมน์  
และแถว จะคำนวณตามขนาดความกว้าง และความสูงของข้อมูลตัวอักษรที่ใช้) จำนวนตัวอักษรสูงสุดที่  
ต้องการจะรับ จะรับข้อความภาษาไทย หรือภาษาอังกฤษ (สำหรับแป้นที่ใช้ในการเปลี่ยน ระหว่าง  
ภาษาไทย และภาษาอังกฤษคือแป้น '~') และสีพื้นของบริเวณที่จะรับข้อความตามลำดับ

ตัวอย่างการใช้งาน

```
text_class text_obj;
unsigned char *string;
string=text_obj.read_str( 5, 2, 50,'E', YELLOW )
```

```
unsigned char *read_eng_str( int column, int row, int count,
                             int b_color )
```

ใช้สำหรับอ่านข้อความเฉพาะภาษาอังกฤษ (ดู read\_str() )

```
void readfont( void )
```

ใช้สำหรับอ่านแฟ้มชื่อข้อมูลอักขระจากแฟ้มข้อมูล "NORMAL.FON" และ "ITALIC.FON"

ตัวอย่างการใช้งาน

```
text_class text_obj;
text_obj.readfont();
```

```
void outthaixy( int column, int row, unsigned char *content )
```

ใช้สำหรับแสดงข้อความทั้งภาษาไทย และภาษาอังกฤษบนจอภาพในตำแหน่งที่กำหนดอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งที่ต้องการจะแสดงข้อความ (พิจารณาเป็นคอลัมน์ และแถวโดยตำแหน่งของคอลัมน์ และแถว จะคำนวณตามขนาดความกว้าง และความสูงของข้อมูลตัวอักษรที่ใช้) และข้อความที่ต้องการแสดงบนจอภาพ

ตัวอย่างการใช้งาน

```
text_class text_obj;
text_obj.outthaixy( 5, 10, "ทดสอบ" );
```

```
void setstyle( int style )
```

ใช้สำหรับเลือกรูปแบบของตัวอักษรที่ต้องการจะแสดงบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ รูปแบบของตัวอักษรที่ต้องการโดยรูปแบบของตัวอักษรที่ได้จัดเตรียมไว้ได้ กำหนดเป็นแมโครไว้ในแฟ้มข้อมูล "text.hpp" ซึ่งสามารถแสดงได้ดังต่อไปนี้

- 1) ITALIC หมายถึงตัวเอียง
- 2) ENLARGE หมายถึงตัวขยาย
- 3) BOLD หมายถึงตัวหนา
- 4) UNDER\_LINE หมายถึงตัวขีดเส้นใต้

```
void clear_style( int style )
```

ใช้สำหรับยกเลิกรูปแบบตัวอักษรที่ไม่ต้องการ (ดู setstyle())

#### 8. คลาสสำหรับการจัดการปุ่มเลือก

คลาสนี้เป็นดีโพรไฟคลาสของคลาส text\_class

หน้าที่ : จัดการปุ่มเลือก

ชื่อคลาส : button\_class

แฟ้มข้อมูลที่กำหนดนิยามของคลาส : button.hpp

วิธีการ :

```
void draw()
```

ใช้สำหรับการวาดปุ่มเลือกลงบนจอภาพ

ตัวอย่างการใช้งาน

```
button_class button_obj;
button_obj.draw;
```

```
void create( int pt_x1, int pt_y1, int pt_x2, int pt_y2, int text_x, int text_y, char* text,
            int bor_clr = BLACK, int pan_clr = LIGHTGRAY, int txt_clr = BLACK )
```

ใช้สำหรับการสร้างปุ่มเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้จะสรุปได้ดังนี้ pt\_x1, pt\_y1, pt\_x2, pt\_y2 จะเป็นบริเวณของปุ่มเลือกบนจอภาพ (กำหนดเป็นพิกเซล) text\_x, text\_y จะเป็นตำแหน่งเริ่มต้นของข้อความที่จะแสดงในปุ่มเลือก (กำหนดเป็นพิกเซล) text หมายถึงข้อความที่จะแสดงในปุ่มเลือก bor\_clr หมายถึงสีขอบของปุ่มเลือก pan\_clr หมายถึงสีของปุ่มเลือก txt\_clr หมายถึงสีของตัวอักษร ที่แสดงในปุ่มเลือก

ตัวอย่างการใช้งาน

```
button_class button_obj;
button_obj.create( 50, 0, 100, 100, 56, 20, BLACK, LIGHTGRAY, BLACK,
                  "ทดสอบ" );
```

```
void clear_button( int pt_x1, int pt_y1, int pt_x2, int pt_y2 )
```

ใช้สำหรับลบปุ่มเลือกออกจากจอภาพ (ดู create\_button())

```
void push_button( void )
```

ใช้สำหรับทำให้ปุ่มมีการเคลื่อนไหวเมื่อมีการกดปุ่ม

ตัวอย่างการใช้งาน

```
button_class button_obj;
button_obj.push_button();
```

```
void set_border_clr( int bor_clr )
```

ใช้สำหรับเปลี่ยนสีขอบของปุ่มเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ สีของขอบที่ต้องการเปลี่ยน

ตัวอย่างการใช้งาน

```
button_class button_obj;
button_obj.set_border_clr( BLACK );
```

```
void set_panel_clr( int pan_clr )
```

ใช้สำหรับเปลี่ยนสีของปุ่มเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือสีที่ต้องการเปลี่ยน (ดู set\_border\_clr)

```
void set_text_clr( int txt_clr )
```

ใช้สำหรับเปลี่ยนสีของตัวอักษรที่แสดงบนปุ่มเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ สีของตัวอักษร (ดู set\_bodr\_clr )

```
void set_label( char *text )
```

ใช้สำหรับเปลี่ยนข้อความที่แสดงบนปุ่มเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ข้อความที่ต้องการจะแสดงบนปุ่มเลือก

```
int button_hit( int mouse_x, int mouse_y )
```

ใช้สำหรับตรวจสอบดูว่ามีการกดปุ่มเลือกหรือไม่ ถ้ามีการกดปุ่มเลือกจะส่งค่า 1 กลับไป ถ้าไม่มีการกดปุ่มเลือกจะส่งค่า 0 กลับไป อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของตัวชี้ (ระบุเป็นพิกเซล)

ตัวอย่างการใช้งาน

```
button_class button_obj;
if( button_obj.button_hit( 50, 50 ) )
{
    ...
}
```

#### 9. คลาสเกี่ยวกับกล่องคำตอบ

คลาสนี้เป็นดีไวส์คลาสของคลาส text\_class

หน้าที่ : จัดการกล่องคำตอบ

ชื่อคลาส : dialog\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : dialog.hpp

วิธีการ :

```
void create_dialog( int x1, int y1, int x2, int y2, int t1x,
    int t1y, char *string1, int t2x = - 1,
    int t2y = -1, char *string2 = NULL,
    int tcolor = BLACK, int bcolor = BLACK,
    int pcolor = LIGHTCYAN )
```

ใช้สำหรับการสร้างกล่องคำโต้ตอบ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ แสดงได้ดังต่อไปนี้ x1, y1, x2, y2 คือบริเวณที่จะสร้างกล่องคำโต้ตอบบนจอภาพ (กำหนดเป็นพิกเซล) t1x, t1y ตำแหน่งเริ่มต้นของข้อความข้อความแรกที่ต้องการแสดงในกล่องคำโต้ตอบ (กำหนดเป็นคอลัมน์ และแถว โดยคำนวณจากความกว้าง และความสูงของข้อมูลตัวอักษรที่ใช้) string1 คือข้อความข้อความแรกที่จะแสดงบนจอภาพ สำหรับ t2x, t2y และ string2จะเป็นตำแหน่งของข้อความข้อความที่สอง และข้อความข้อความที่สองตามลำดับ ซึ่งข้อความที่สองนี้จะมี หรือไม่มีก็ได้ ส่วน tcolor, bcolor และ pcolor จะเป็นสีของตัวอักษร สีของขอบของกล่องคำโต้ตอบ และสีของกล่องคำโต้ตอบตามลำดับ โดยสีต่าง ๆ ได้กำหนดค่าโดยปริยายไว้ให้แล้ว

หมายเหตุ ในกรณีที่ต้องการจะกำหนดสีต่าง ๆ เอง และไม่ต้องการให้มีข้อความข้อความที่สอง จะต้องกำหนดค่าตำแหน่งของข้อความที่สอง และค่าของข้อความที่สอง ให้มีค่าเท่ากับค่าที่กำหนดโดยปริยาย แล้วจึงกำหนดสีต่าง ๆ ได้

ตัวอย่างการใช้งาน

```
 dialog_class dialog_obj;
dialog_obj.create_dialog( 0, 0, 200, 200, 1, 1, "ทดสอบกล่องคำโต้ตอบ" );
```

```
int draw_dialog( void )
```

ใช้สำหรับการวาดกล่องคำโต้ตอบลงบนจอภาพ โดยจะเก็บข้อมูลของจอภาพบริเวณที่จะวาดกล่องคำโต้ตอบไว้ในหน่วยความจำ ในกรณีที่หน่วยความจำไม่พอสำหรับการจัดเก็บข้อมูลของจอภาพ จะไม่วาดกล่องคำโต้ตอบให้ และส่งค่า -1 กลับมา

ตัวอย่างการใช้งาน

```
dialog_class dialog_obj;
if( dialog_obj.draw_dialog() == -1 )
{
    ...
}
```

```
void close_dialog( void )
```

ใช้สำหรับลบกล่องคำโต้ตอบออกจากจอภาพ  
และเขียนข้อมูลของจอภาพเดิมที่ได้เก็บไว้ในหน่วยความจำในตอนวาดกล่องคำโต้ตอบลงบนจอภาพ  
ตัวอย่างการใช้งาน

```
dialog_class dialog_obj;  
dialog_obj.close_dialog();
```

#### 10. คลาสเกี่ยวกับการจัดการสโกรลบาร์

คลาสนี้เป็นดีโพรไฟคลาสของคลาส viewport\_class

หน้าที่ : จัดการสโกรลบาร์

ชื่อคลาส : scrollbar\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : scribar.hpp

วิธีการ :

```
void draw()
```

ใช้สำหรับการวาดสโกรลบาร์ลงบนจอภาพ

ตัวอย่างการใช้งาน

```
scrollbar_class scrollbar_obj;  
scrollbar_obj.draw();
```

```
void init( int pt_x, int pt_y, int size, int th_size_x, int th_size_y, int orientation,
          int bor_clr = WHITE, int pan_clr = LIGHTBLUE, int endp_clr = DARKGRAY,
          int thum_clr = DARKGRAY, int arr_clr = BLACK )
```

ใช้สำหรับการสร้างสโกลบาร์ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ สามารถแสดงได้ดังต่อไปนี้  
 pt\_x, pt\_y คือตำแหน่งเริ่มต้นของสโกลบาร์บนจอภาพ (กำหนดเป็นพิกเซล) size คือความยาว  
 (หรือความสูงของสโกลบาร์ ในกรณีที่เป็นสโกลบาร์ในแนวตั้ง (Vertical Scrollbar)) th\_size\_x,  
 th\_size\_y เป็นขนาดของปุ่มแพด (Thumbpad) orientation เป็นตัวบอกว่าสโกลบาร์นี้เป็นสโกลบาร์  
 ในแนวนอน (Horizontal Scrollbar) หรือสโกลบาร์ในแนวตั้ง โดยค่าของ orientation ใช้แม่โครที่  
 กำหนดไว้ในแฟ้มข้อมูล "graphics.h" คือ

- 1) HORIZ\_DIR หมายถึงสโกลบาร์ในแนวนอน (Horizontal Scrollbar)
- 2) VERT\_DIR หมายถึงสโกลบาร์ในแนวตั้ง

bor\_clr เป็นสีของขอบของสโกลบาร์ pan\_clr เป็นสีของสโกลบาร์ endp\_clr เป็นสีของเอนด์แพด  
 (Endpad) ของสโกลบาร์ thum\_clr เป็นสีของปุ่มแพดของสโกลบาร์ และ arr\_clr เป็นสีของลูกศรที่อยู่  
 บนเอนด์แพดของสโกลบาร์

ตัวอย่างการใช้งาน

```
scrollbar_class scrollbar_obj;
scrollbar_obj.init( 50, 50, 400, 16, 16, HORIZ_DIR );
```

ศูนย์วิทยพัทยาการ  
 จุฬาลงกรณ์มหาวิทยาลัย

```
hit_type scroll_hit( int mouse_x, int mouse_y )
```

ใช้สำหรับตรวจสอบว่ามีการเรียกใช้สโกลบาร์หรือไม่สำหรับค่าที่จะส่งกลับจากวิธีการนี้แสดงได้ดังนี้

- 1) NO\_HIT หมายถึงไม่มีการเรียกใช้สโกลบาร์
- 2) LEFT หมายถึงมีการเรียกใช้สโกลบาร์โดยกดเอนต์แปดทางซ้ายของสโกลบาร์ในแนวนอน
- 3) RIGHT หมายถึงมีการเรียกใช้สโกลบาร์โดยกดเอนต์แปดทางขวาของสโกลบาร์ในแนวนอน
- 4) UP หมายถึงมีการเรียกใช้สโกลบาร์โดยกดเอนต์แปดทางด้านบนของสโกลบาร์ในแนวตั้ง
- 5) DOWN หมายถึงมีการเรียกใช้สโกลบาร์โดยกดเอนต์แปดทางด้านล่างของสโกลบาร์ในแนวตั้ง
- 6) HBAR หมายถึงมีการเรียกใช้สโกลบาร์โดยกดที่ตัวสโกลบาร์ในแนวนอน
- 7) VBAR หมายถึงมีการเรียกใช้สโกลบาร์โดยกดที่ตัวสโกลบาร์ในแนวตั้ง

อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของตัวชี้

ตัวอย่างการใช้งาน

```
scrollbar_class scrollbar_obj;
switch( scrollbar_obj.scroll_hit( 50, 50 ) )
{
    case NOHIT:
        ...
        break;
    case LEFT:
        ...
        break;
}
```

```
int get_position_thumbpad( void )
```

ใช้สำหรับรายงานตำแหน่งของทัมแพด

ตัวอย่างการใช้งาน

```
scrollbar_class scrollbar_obj;
int pos;
pos = scrollbar_obj.get_position();
```

หมายเหตุ โดยทั่วไปจะใช้วิธีการนี้เมื่อต้องการจะลบสโกลบาร์จากหน้าจอ และต้องการจะให้ทัมแพดอยู่ในตำแหน่งเดิมเมื่อมีการวาดสโกลบาร์ลงบนจอภาพอีกครั้ง เนื่องจากวิธีการ draw() ในการวาดสโกลบาร์ลงบนจอภาพ จะกำหนดให้ทัมแพดเริ่มที่ตำแหน่งซ้ายสุดของสโกลบาร์ในแนวนอนหรือบนสุดของสโกลบาร์ในแนวตั้ง ดังนั้นวิธีแก้ก็คือใช้วิธีการ get\_position() นี้เก็บตำแหน่งของทัมแพดไว้ จากนั้นจึงเรียกใช้วิธีการ set\_position\_thumbpad() (ดู set\_position\_thumbpad())



```
int in_scrollbar( int cur_x, int cur_y )
```

ใช้สำหรับตรวจสอบว่าตัวชี้ตำแหน่งอยู่ในบริเวณสโกลบาร์หรือไม่ ถ้าตัวชี้ตำแหน่งอยู่ในบริเวณสโกลบาร์ จะส่งค่า 1 กลับไป มิฉะนั้นจะส่งค่า 0 กลับไป

ตัวอย่างการใช้งาน

```
scrollbar_class scrollbar_obj;
if( scrollbar_obj.in_scrollbar( 50, 50 ) )
{
    ...
}
```

```
void set_step( int st )
```

ใช้สำหรับกำหนดว่าจะให้ทิมแพดเลื่อนไปเท่าไรเมื่อมีการเรียกใช้สโกลบาร์ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ค่าที่ต้องการให้ทิมแพดเลื่อนไปเมื่อมีการเรียกใช้สโกลบาร์ ค่าอาร์กิวเมนต์นี้จะเกิดจากการคำนวณ โดยใช้ความกว้างของรูปภาพ หรือข้อมูล ทารด้วยความยาว (ใช้ความสูงในกรณีเป็นสโกลบาร์ในแนวตั้ง) ของสโกลบาร์

ตัวอย่างการใช้งาน

```
scrollbar_class scrollbar_obj;
scrollbar_obj.set_step( 5 );
```

```
int get_step( void )
```

ใช้สำหรับรายงานค่าที่กำหนดให้ทิมแพดเคลื่อนที่ไปเมื่อมีการเรียกใช้สโกลบาร์ (ดู set\_step())

ตัวอย่างการใช้งาน

```
scrollbar_class scrollbar_obj;
int step;
step = scrollbar_obj.get_step();
```

```
void set_position_thumbpad( int pos )
```

ใช้สำหรับกำหนดทิมแพดให้เลื่อนไปในตำแหน่งที่ต้องการ โดยทั่วไปจะใช้ร่วมกับ get\_position\_thumbpad() (ดู get\_position\_thumbpad())

## 11. คลาสสำหรับการวาดจุดลงบนจอภาพ

คลาสนี้เป็นดีโพรไฟคลาสของคลาส figure\_class

หน้าที่ : วาดจุดลงบนจอภาพ

ชื่อคลาส : point\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

วิธีการ :

```
void init_point(int x, int y, int clr, int old_mode, int sz )
```

ใช้สำหรับกำหนดข้อมูลที่จำเป็นสำหรับการวาดจุดลงบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งที่ต้องการจะวาดจุด สีของจุด วิธีที่ใช้ในการวาดภาพ และขนาดของจุดสำหรับวิธีที่ใช้ในการวาดภาพจะใช้เมโครที่อยู่ในแฟ้มข้อมูล "graphics.h" ดังนี้

1) COPY\_PUT หมายถึงให้วาดรูปลงบนจอภาพจริง ๆ

2) XOR\_PUT หมายถึงให้วาดภาพเพื่อดูโครงร่างของภาพเท่านั้น ยังไม่วาดลงจอภาพจริง ๆ

ตัวอย่างการใช้งาน

```
point_class point_obj;
```

```
point_obj.init_point( 50, 50, YELLOW, COPY_PUT, 1 )
```

สำหรับวิธีการวาดรูปลงบนจอภาพ ดู 18.

## 12. คลาสสำหรับวาดเส้นตรง

คลาสนี้เป็นดีโพรไฟคลาสของคลาส point\_class

หน้าที่ : วาดเส้นตรงบนจอภาพ

ชื่อคลาส : line\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

วิธีการ :

```
void init_line( int x1, int y1, int x2, int y2, int clr, int old_mode, int sz )
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับการเส้นตรงบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ก็คือ จุดตั้งต้น และสิ้นสุดของเส้นตรง สีของเส้น วิธีที่ใช้ในการวาดภาพ (ดู 11.) และขนาดของเส้น

ตัวอย่างการใช้งาน

```
line_class line_obj;
```

```
line_obj.init_line( 50, 50, 100, 100, YELLOW, COPY_PUT, 1 );
```

สำหรับวิธีการวาดรูปลงบนจอภาพ ดู 18.

## 13. คลาสสำหรับการวาดสี่เหลี่ยม

คลาสนี้เป็นดีโพรไฟคลาสของคลาส line\_class

หน้าที่ : วาดรูปสี่เหลี่ยมบนจอภาพ

ชื่อคลาส : rect\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

วิธีการ :

```
void init_rect( int x1, int y1, int x2, int y2, int clr, int old_mode, int sz, int fill )
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับการวาดรูปสี่เหลี่ยมบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ก็คือ จุดขอบบนซ้าย และล่างขวาของรูปสี่เหลี่ยม สีของรูป วิธีการวาดภาพ (ดู 11.) ขนาดของกรอบรูป และ fill จะเป็นตัวที่บอกว่าจะวาดรูปสี่เหลี่ยมแบบโปร่ง หรือรูปสี่เหลี่ยมแบบทึบ ถ้า fill มีค่าเป็น 0 จะเป็นสี่เหลี่ยมแบบโปร่ง ถ้า fill มีค่าเป็น 1 จะเป็นสี่เหลี่ยมแบบทึบ

ตัวอย่างการใช้งาน

```
rect_class rect_obj;
```

```
rect_obj.init_rect( 50, 50, 100, 100, YELLOW, COPY_PUT, 1, 1 );
```

สำหรับวิธีการวาดรูปลง บนจอภาพ ดู 18.

## 14. คลาสสำหรับการสร้างรูปวงรี

คลาสนี้เป็นดีโพรไฟคลาสของคลาส rect\_class

หน้าที่ : วาดรูปวงรีบนจอภาพ

ชื่อคลาส : ellipse\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

วิธีการ :

```
void init_ellipse( int x,int y,int r1, int r2, int clr, int old_mode, int sz, int fill )
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับการวาดรูปวงรีบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือจุดศูนย์กลางของวงรี รัศมีตามแนวแกน x และ แกน Y สีของรูป วิธีการวาดภาพ (ดู 11.) ขนาดของเส้นกรอบ และตัวที่บอกว่าจะวาดรูปวงรีแบบโปร่ง หรือวงรีแบบทึบ (ดู 13.) ในกรณีที่ต้องการจะวาดวงกลมก็ให้ค่ารัศมีตามแนวแกน X และแกน Y มีค่าเท่ากัน

ตัวอย่างการใช้งาน

```
ellipse_class ellipse_obj;
```

```
ellipse_obj.init_ellipse( 50, 50, 30, 40, YELLOW, COPY_PUT, 1, 0 );
```

สำหรับวิธีการวาดรูปลงบนจอภาพ ดู 18.

## 15. คลาสสำหรับการวาดเส้นโค้ง

คลาสนี้เป็นดีโพรไฟคลาสของคลาส line\_class

หน้าที่ : วาดรูปเส้นโค้งบนจอภาพ

ชื่อคลาส : curve\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

วิธีการ :

```
void init_curve( int start_x, int start_y, int ctrl_x, int ctrl_y, int end_x, int end_y, int old_mode,
                int clr, int sz );
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับการวาดรูปเส้นโค้งบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ก็คือ จุดควบคุมการวาดเส้นโค้ง 3 จุด ได้แก่ จุด (start\_x, start\_y) จุด (ctrl\_x, ctrl\_y) และ จุด (end\_x, end\_y) ตามลำดับ วิธีในการวาดภาพ (ดู 11.) สีของเส้นโค้ง และขนาดของเส้นโค้ง

ตัวอย่างการใช้งาน

```
curve_class curve_obj;
```

```
curve_obj.init_curve( 50 , 50, 100, 80, 200, 200, COPY_PUT, YELLOW, 1 );
```

สำหรับวิธีการวาดรูปลงบนจอภาพ ดู 18.

## 16. คลาสสำหรับการจัดการบริเวณที่ผู้ใช้เลือกบนจอภาพ

คลาสนี้เป็นดีโพรไฟคลาสของคลาส rect\_class

หน้าที่ : วาดเส้นกรอบบริเวณที่ผู้ใช้เลือกบนจอภาพ และ เก็บจุดขอบบนซ้าย และล่างขวาของบริเวณที่ผู้ใช้เลือก

ชื่อคลาส : selection\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

โครงสร้างข้อมูลที่กำหนดในคลาสนี้:

```
struct area
{
    int left, top, right, bottom;
};
```

โดยตัวแปร left, top จะเก็บจุดขอบบนด้านซ้าย และ right, bottom จะเก็บจุดขอบล่างด้านขวาของบริเวณที่ผู้ใช้เลือก

วิธีการ

```
area get_area( void )
```

ใช้สำหรับรายงานค่าจุดขอบบนด้านซ้าย และจุดขอบล่างด้านขวาของบริเวณที่ผู้ใช้เลือก ตัวอย่างการใช้งาน

```
selection_class select_obj;
area area_var;
area_var = select_obj.get_area();
```

```
int in_select_area( int curx, int cury )
```

ใช้สำหรับตรวจสอบว่าขณะนั้นตัวชี้ตำแหน่งอยู่ในบริเวณที่เลือกไว้หรือไม่ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของตัวชี้บนจอภาพ (เป็นพิกเซล) โดยวิธีการนี้จะส่งค่ากลับเป็น 1 ถ้าตัวชี้ตำแหน่งอยู่ในบริเวณที่ผู้ใช้เลือก มิฉะนั้นจะส่งค่ากลับเป็น 0

ตัวอย่างการใช้งาน

```
selection_class select_obj;
if( select_obj.in_select_area( 50, 50 ) )
{
    ...
}
```

สำหรับวิธีการวาดรูปลงบนจอภาพ ดู 18.

### 17. คลาสสำหรับการพ่นสี (Spray Can)

คลาสนี้เป็นดีโพรไฟคลาสของคลาส point\_class

หน้าที่ : พ่นสีลงบนจอภาพ

ชื่อคลาส : airbrush\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

วิธีการ :

สำหรับวิธีการวาดรูปลงบนจอภาพ ดู 18.

### 18. วิธีการสำหรับการวาดรูปกราฟิกต่าง ๆ ลงบนจอภาพ

เนื่องจากการต้องการให้การวาดภาพลงบนจอภาพมีความสะดวก โดยใช้คำสั่งที่เหมือนกันในการวาดภาพต่าง ๆ จึงได้จัดทำฟังก์ชันสำหรับการวาดภาพไว้ดังนี้

1) void draw\_outline( figure \*info\_holder);

ใช้สำหรับวาดภาพเพื่อดูโครงร่างของภาพ (ยังไม่วาดลงจอภาพจริง ๆ)

2) void draw\_picture( figure \*info\_holder);

ใช้สำหรับการวาดภาพบนจอภาพ

โดยการเรียกใช้ฟังก์ชันทั้งสอง จะต้องประกาศตัวแปรของคลาสรูปภาพต่าง ๆ ให้เป็นตัวแปรแบบพอยน์เตอร์ เช่นถ้าต้องการจะวาดภาพเส้นตรงจะมีวิธีการดังนี้

```
line_class *line_obj = new line_class;
```

```
line_obj->init_line( 50, 50, 100, 100, YELLOW, COPY_PUT, 1);
```

```
draw_picture( line_obj );
```

### 19. คลาสสำหรับการระบายสีรูปภาพ

คลาสนี้เป็นดีโพรไฟคลาสของคลาส viewport\_class

หน้าที่ : วาดรูปเส้นโค้งบนจอภาพ

ชื่อคลาส : roller\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : drawtool.hpp

วิธีการ :

```
void jfill( int x, int y )
```

ใช้สำหรับการระบายสีลงในรูปภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือจุดที่จะเริ่มระบายสี (ระบุเป็นพิกเซล)

ตัวอย่างการใช้งาน

```
roller_class roller_obj;
```

```
roller_obj.jfill( 50, 50 );
```

## 20. คลาสสำหรับจัดการรายการเลือกของรายการคำสั่ง

คลาสนี้เป็นดีไรฟ์คลาสของคลาส viewport\_class

หน้าที่ : จัดการรายการเลือกของรายการคำสั่ง

ชื่อคลาส : gmenu\_item\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : menuhead.hpp

วิธีการ :

```
void init( int num )
```

ใช้สำหรับกำหนดค่าเริ่มต้นที่จำเป็นสำหรับรายการเลือกของรายการคำสั่ง อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จำนวนรายการคำสั่งทั้งหมดในรายการเลือกนี้

ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
```

```
menu_obj.init( 5 );
```

```
void create_menu( int item_num, char *name, int x1, int y1, int size_x, int act, char acdisp,
                 int apos, char actv, int sub, void(*f)() = NULL, int color1 = BLACK,
                 int color2 = RED )
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับรายการคำสั่งแต่ละตัว อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้แสดงได้ดังนี้ item\_num คือรายการคำสั่งตัวที่ item\_num name หมายถึงข้อความของรายการคำสั่ง x1, y1 คือตำแหน่งเริ่มต้นที่จะแสดงรายการคำสั่ง (กำหนดเป็นคอลัมน์ และแถวคำนวณจากความกว้าง และความสูงของข้อมูลตัวอักษร) act เป็นแฟล็ก (Flag) ที่จะบอกว่าเมื่อเลือกรายการคำสั่งนี้แล้วจะมีการทำงานหรือไม่ acdisp คือตัวอักษรที่แสดงให้เห็นบนจอภาพว่าเป็นแอ็กทิเวชันคีย์ (Activation key) apos ตำแหน่งที่จะแสดงแอ็กทิเวชันคีย์บนจอภาพ actv แอ็กทิเวชันคีย์ที่ตรงกับแป้นพิมพ์ภาษาอังกฤษ sub เป็นแฟล็กที่บอกว่า ยังมีรายการเลือกย่อยของรายการคำสั่งนี้อีกหรือไม่ ถ้า sub มีค่าเป็น 1 แสดงว่ามีรายการเลือกย่อย f จะเป็นฟังก์ชันที่จะเรียกใช้เมื่อมีการเลือกรายการเลือกตัวนี้ color1 เป็นสีของตัวอักษร ที่เป็นข้อความ color2 เป็นสีของตัวอักษรที่เป็นแอ็กทิเวชันคีย์

หมายเหตุ ความแตกต่างระหว่าง actvdisp และ actv จะขอยกตัวอย่างดังนี้ เช่นต้องการจะแสดงแอ็กทิเวชันคีย์เป็นตัว 'ก' ก็กำหนดให้ actvdisp เป็นตัว 'ก' แต่ตัว 'ก' ตรงกับตัวอักษรภาษาอังกฤษบนแป้นพิมพ์คือ 'd' ดังนั้นจะกำหนด actv เป็น 'd'

ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
```

```
menu_obj.init( 5 );
```

```
menu_obj.create_menu( 0, "เมนูหลัก", 2, 3, 20, 1, 'l', 1, 'g', 1, do_file );
```

```
...
```

```
void create_menus(int dir, int color1 = YELLOW, int color2 = WHITE, int color3 = BLACK )
```

ใช้สำหรับสร้างรายการเลือกของรายการคำสั่ง อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้แสดงได้ดังนี้ dir จะเป็นตัวบอกว่าเป็นรายการเลือกตามแนวนอนหรือแนวตั้ง โดยค่าที่จะส่งผ่านอาร์กิวเมนต์นี้จะมีได้ 2 ค่าดังนี้

- 1) HORIZ\_DIR หมายถึงรายการเลือกตามแนวนอน (รายการหลัก)
- 2) VERT\_DIR หมายถึงรายการเลือกตามแนวตั้ง (รายการย่อยของรายการหลัก)

ส่วนอาร์กิวเมนต์ color1, color2 และ color3 เป็นสีของแถบสว่าง สีพื้นของรายการเลือก และสีของตัวอักษรตามลำดับ ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
menu_obj.init( 5 );
menu_obj.create_menu( 0, "เมนูหลัก1", 2, 3, 20, 1, 'l', 1, 'g', 1, do_file );
menu_obj.create_menu( 1, "เมนูหลัก2", 23, 3, 20, 1, 'm', 2, 'm', 1, do_file1 );
...
menu_obj.create_menus( HORIZ_DIR );
```

```
int draw_menu( void )
```

ใช้สำหรับการแสดงรายการเลือกของรายการคำสั่งบนจอภาพ โดยจะเก็บข้อมูลของจอภาพที่อยู่ในบริเวณที่จะแสดงรายการเลือกไว้ในหน่วยความจำ ในกรณีที่หน่วยความจำไม่พอที่จะเก็บบริเวณของจอภาพที่จะแสดงรายการเลือก จะส่งค่า -1 กลับมา และไม่แสดงรายการเลือก

ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
menu_obj.init( 5 );
menu_obj.create_menu( 0, "เมนูหลัก1", 2, 3, 20, 1, 'l', 1, 'g', 1, do_file );
menu_obj.create_menu( 1, "เมนูหลัก2", 2, 3, 20, 1, 'l', 1, 'g', 1, do_file );
...
menu_obj.create_menus( HORIZ_DIR );
if( menu_obj.draw_menu() == -1 )
{
...
}
```



```
void erase_menu( void )
```

ใช้สำหรับลบรายการเลือกของรายการคำสั่งออกจากจอภาพ

ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
....
menu_obj.erase_menu();
```

```
void set_pos( int i )
```

ใช้สำหรับกำหนดตำแหน่งของสมาชิกของรายการเลือกที่จะแสดงแถบสว่าง อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ สมาชิกของรายการเลือกที่ต้องการจะให้มีความสว่าง

ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
...
menu_obj.set_pos( 0 );
```

```
void run_func( int i )
```

ใช้สำหรับเรียกฟังก์ชันที่ถูกกำหนดให้ทำงานเมื่อสมาชิกของรายการเลือกถูกเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ สมาชิกของรายการเลือกที่ถูกเลือก

ตัวอย่างการใช้งาน

```
g_menu_class menu_obj;
...
menu_obj.run_func( 0 );
```

```
int in_menu( int cur_x, int cur_y )
```

ใช้สำหรับตรวจสอบว่าตัวชี้ตำแหน่งอยู่ในบริเวณรายการเลือกหรือไม่ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ก็คื ตำแหน่งของตัวชี้ตำแหน่งบนจอภาพ (เป็นพิกเซล)

ตัวอย่างการใช้งาน

```
gmenu_class mnu_obj;
...
if( menu_obj.in_menu( 50, 50 ) )
{
...
}
```

```
void set_item_active( int num, int act )
```

ใช้สำหรับกำหนดให้สมาชิกของรายการเลือกพร้อมที่จะทำงาน หรือไม่ทำงานเมื่อถูกเลือกอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ สมาชิกตัวที่ต้องการจะทำงานด้วย และค่าที่กำหนดว่าพร้อมจะทำงานหรือไม่ทำงานเมื่อถูกเลือก โดยถ้าต้องการให้ทำงานเมื่อถูกเลือก ก็กำหนดค่า act ให้เป็น 1 ถ้าไม่ต้องการจะให้ทำงาน ก็กำหนดค่า act ให้เป็น 0

ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
...
menu_obj.set_item_active( 0, 1 );
```

```
void set_item_color( int num, int color )
```

ใช้สำหรับกำหนดสีของข้อความของสมาชิกของรายการเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ สมาชิกของรายการเลือกที่ต้องการทำงานด้วย และสีของตัวอักษรที่เป็นข้อความ

ตัวอย่างการใช้งาน

```
gmenu_class menu_obj;
...
menu_obj.set_item_color( 0, YELLOW );
```

## 21. คลาสสำหรับจัดการรายการเลือกสีของเส้นกรอบ และสีสำหรับการระบายภาพ

คลาสนี้เป็นดีโพรไฟคลาสของคลาส gmenu\_class

หน้าที่ : จัดการรายการเลือกสี

ชื่อคลาส : clrmenu\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : menuhead.hpp

วิธีการ :

```
void init( int num )
```

ใช้สำหรับกำหนดค่าเริ่มต้นที่จำเป็นสำหรับรายการเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จำนวนรายการคำสั่งทั้งหมดในรายการเลือกนี้

ตัวอย่างการใช้งาน

```
clrmenu_class color_obj;
color_obj.init( 5 );
```

```
void create_menu( int num, int color, int start_x, int start_y, int end_x, int end_y,
                 int bor_clr = BLACK )
```

ใช้สำหรับสร้างสมาชิกแต่ละตัวของรายการเลือกสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้แสดงได้ดังนี้ num หมายถึงสมาชิกตัวที่ num ของรายการเลือก color หมายถึงสีที่เป็นค่าของสมาชิกตัวนั้น start\_x, start\_y, end\_x, end\_y เป็นตำแหน่งที่ปรากฏบนจอภาพของสมาชิกตัวนั้น โดยจะแสดงสมาชิกแต่ละตัวเป็นรูปสี่เหลี่ยมโดย start\_x, start\_y จะเป็นจุดขอบบนด้านซ้าย และ end\_x, end\_y จะเป็นจุดมุมล่างขวา bor\_clr หมายถึงสีของขอบ

ตัวอย่างการใช้งาน

```
clrmenu_class color_obj;
color_obj.init( 5 );
color_obj.create_menu( 0, BLACK, 50, 50, 80, 80, BLACK );
...
```

```
void create_menus( int left, int top, int right, int bottom, int bor_clr = BLACK,
                  int p_clr = WHITE )
```

ใช้สำหรับสร้างรายการเลือกสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือตำแหน่งของรายการสีบนจอภาพ โดยจะพิจารณาเป็นรูปสี่เหลี่ยม โดย left, top จะเป็นจุดมุมบนซ้าย ส่วน right, bottom จะเป็นมุมล่างขวา bor\_clr และ p\_clr จะเป็นสีขอบของรายการเลือก และสีพื้นของรายการเลือกตามลำดับ

ตัวอย่างการใช้งาน

```
clrmenu_class color_obj;
color_obj.init( 5 );
color_obj.create_menu( 0, BLACK, 50, 50, 80, 80, BLACK );
color_obj.create_menu( 1, BLACK, 100, 50, 130, 80, BLACK );
...
color_obj.create_menus( 0, 400, 400, 480 );
```

```
void draw_menu( int num )
```

ใช้สำหรับการแสดงรายการสีบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือสมาชิกที่ต้องการจะให้เป็นที่ใช้ในการวาดภาพ หลังจากการวาดรายการสีบนจอภาพ

ตัวอย่างการใช้งาน

```
clrmenu_class color_obj;
...
color_obj.draw_menu( 0 );
```

```
int color_obj.get_item_hit( int xcur, int ycur )
```

ใช้สำหรับตรวจสอบว่ามีการเลือกสีใดจากรายการสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของตัวชี้ตำแหน่งบนจอภาพ

ตัวอย่างการใช้งาน

```
clrmenu_class color_obj;
int i;
...
i = color_obj.get_item_hit( 50, 50 )
```

## 22. คลาสสำหรับการจัดการรายการเลือกขนาดของเส้น

คลาสนี้เป็นดีโพรไฟคลาสของคลาส clrmenu\_class

หน้าที่ : จัดการรายการเลือกของขนาดของเส้น

ชื่อคลาส : linemenu\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : menuhead.hpp

วิธีการ :

```
void init( int num )
```

ใช้สำหรับกำหนดค่าเริ่มต้นที่จำเป็นสำหรับรายการเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จำนวนรายการคำสั่งทั้งหมดในรายการเลือกนี้

ตัวอย่างการใช้งาน

```
linemenu_class line_obj;
line_obj.init( 5 );
```

```
void create_menu( int num, int start_x, int start_y, int end_x, int end_y, int sz,
int line_clr = BLACK );
```

ใช้สำหรับสร้างสมาชิกแต่ละตัวของรายการเลือกขนาดของเส้น อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้แสดงได้ดังนี้ num หมายถึงสมาชิกตัวที่ num start\_x, start\_y, end\_x, end\_y เป็นตำแหน่งที่ปรากฏบนจอภาพของสมาชิกตัวนั้น โดยจะแสดงสมาชิกแต่ละตัวเป็นรูปสี่เหลี่ยม start\_x, start\_y จะเป็นจุดมุมบนซ้าย และ end\_x, end\_y จะเป็นจุดมุมล่างขวา sz คือขนาดของเส้น line\_clr หมายถึงสีของเส้นที่จะแสดง

ตัวอย่างการใช้งาน

```
linemenu_class line_obj;
line_obj.init( 5 );
line_obj.create_menu( 0, 50, 50, 50, 58, 1, BLACK );
line_obj.create_menu( 1, 50, 58, 50, 66, 2, BLACK );
...
int get_item_hit( int x_pos, int y_pos )
```

ใช้สำหรับตรวจสอบว่า มีการเลือกสีใดจากรายการสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของตัวชี้ตำแหน่งบนจอภาพ

ตัวอย่างการใช้งาน

```
linemenu_class line_obj;
int i;
...
i = line_obj.get_item_hit( 50, 50 )
```

```
void draw_menu( void );
```

ใช้สำหรับการแสดงรายการเลือกของเส้นบนจอภาพ

ตัวอย่างการใช้งาน

```
linemenu_class line_obj;
...
line_obj.draw_menu();
```

### 23. คลาสสำหรับรายการเลือกกลดลยสำหรับการระบายสี

คลาสนี้เป็นดีโอฟิลคลาสของคลาส clrmenu\_class

หน้าที่ : จัดการรายการเลือกกลดลยสำหรับการระบายสี

ชื่อคลาส : pattmenu\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : menuhead.hpp

วิธีการ :

```
void init( int num )
```

ใช้สำหรับกำหนดค่าเริ่มต้นที่จำเป็นสำหรับรายการเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จำนวนรายการคำสั่งทั้งหมดในรายการเลือกนี้

ตัวอย่างการใช้งาน

```
pattmenu_class patt_obj;
patt_obj.init( 5 );
```

```
void create_menu( int num, int start_x, int start_y, int end_x, int end_y, int f_type,
                 int i = 0, char *h = NULL, int color = YELLOW, int bor_clr = BLACK )
```

ใช้สำหรับสร้างสมาชิกแต่ละตัวของรายการเลือกกลดลายการระบายสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้แสดงได้ดังนี้ num หมายถึงสมาชิกตัวที่ num start\_x, start\_y, end\_x, end\_y เป็นตำแหน่งที่จะแสดงบนจอภาพของสมาชิกตัวนั้น โดยจะแสดงสมาชิกแต่ละตัวเป็นรูปสี่เหลี่ยมโดย start\_x, start\_y จะเป็นจุดมุมบนซ้าย และ end\_x, end\_y จะกำหนดจุดมุมล่างด้านขวา f\_type จะเป็นตัวกำหนดว่าจะใช้ลวดลายของสีที่ตัวแปลโปรแกรม TURBO C++ เตรียมไว้ให้ หรือจะใช้ลวดลายของสีที่ผู้พัฒนาโปรแกรมกำหนดขึ้นเอง โดยถ้าต้องการจะใช้ลวดลายสีของตัวแปลโปรแกรม TURBO C++ ก็กำหนดค่า f\_type เป็น TC\_FILL ถ้าต้องการจะใช้ลวดลายที่กำหนดขึ้นเอง ก็กำหนดค่า f\_type เป็น USER\_FILL สำหรับอาร์กิวเมนต์ i จะขึ้นอยู่กับค่าของ f\_type โดยถ้าค่าของ f\_type เป็น USER\_FILL ค่าของ i จะต้องเป็น 0 เสมอ แต่ถ้าค่า f\_type เป็น TC\_FILL ค่าของ i จะเป็นตัวกำหนดว่าจะใช้ลวดลายของตัวแปลโปรแกรม TURBO C++ ลวดลายใด สำหรับ h จะเป็นตัวชี้ไปยังตำแหน่งที่เก็บข้อมูลลวดลาย ที่ผู้พัฒนาโปรแกรมได้กำหนดไว้ สำหรับ color และ bor\_clr จะเป็นสีที่ใช้ในการระบายลวดลาย และสีขอบของสมาชิกแต่ละตัวของรายการเลือกกลดลาย ตามลำดับ

ตัวอย่างการใช้งาน

```
pattmenu_class patt_obj;
patt_obj.init( 5 );
patt_obj.create_menu( 0, 50, 50, 50, 58, TC_FILL, 1 );
patt_obj.create_menu( 1, 50, 58, 50, 66, USER_FILL, 0, pattern1 );
...
```

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

```
void create_menus( int left, int top, int right, int bottom, int bor_clr = BLACK,
                  int p_clr = WHITE );
```

ใช้สำหรับสร้างรายการเลือกสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของรายการสีบนจอภาพ ซึ่งจะพิจารณาเป็นรูปสี่เหลี่ยม โดย left, top จะเป็นจุดมุมบนซ้าย ส่วน right, bottom จะเป็นมุมล่างขวา bor\_clr และ p\_clr จะเป็นสีขอบของรายการเลือก และสีพื้นของรายการเลือกตามลำดับ

ตัวอย่างการใช้งาน

```
pattmenu_class patt_obj;
patt_obj.init( 5 );
patt_obj.create_menu( 0, 50, 50, 50, 58, TC_FILL, 1 );
patt_obj.create_menu( 1, 50, 58, 50, 66, USER_FILL, 0, pattern1 );
...
patt_obj.create_menus( 0, 400, 400, 480 );
```

```
int get_item_hit( int x_pos, int y_pos )
```

ใช้สำหรับตรวจสอบว่า มีการเลือกสีใดจากรายการสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของตัวชี้ตำแหน่งบนจอภาพ

ตัวอย่างการใช้งาน

```
pattmenu_class patt_obj;
int i;
...
i = patt_obj.get_item_hit( 50, 50 )
```

```
void draw_menu( int num, int color )
```

ใช้สำหรับการแสดงรายการเลือกของขนาดของเส้นบนจอภาพ

ตัวอย่างการใช้งาน

```
pattmenu_class patt_obj;
...
patt_obj.draw_menu();
```

#### 24. คลาสสำหรับรายการเลือกเครื่องมือวาดภาพ

คลาสนี้เป็นดีโพรไฟคลาสของคลาส clrmenu\_class

หน้าที่ : จัดการรายการเลือกเครื่องมือวาดภาพ

ชื่อคลาส : iconmenu\_class

แฟ้มข้อมูลที่กำหนดนิยามของคลาส : menuhead.hpp

วิธีการ :

```
void init( int num )
```

ใช้สำหรับกำหนดค่าเริ่มต้นที่จำเป็นสำหรับรายการเลือก อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้ คือ จำนวนรายการคำสั่งทั้งหมดในรายการเลือกนี้

ตัวอย่างการใช้งาน

```
iconmenu_class icon_obj;
icon_obj.init( 5 );
```

```
void create_menu( int num, int start_x, int start_y, int end_x, int end_y, int f_type,
```

```
int t_clr = WHITE, int bg_clr = LIGHTBLUE, int bor_clr = BLACK );
```

ใช้สำหรับสร้างสมาชิกแต่ละตัวของรายการเลือกวาดลายการระบายสี อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้แสดงได้ดังนี้ num หมายถึงสมาชิกตัวที่ num start\_x, start\_y, end\_x, end\_y เป็นตำแหน่งที่ปรากฏบนจอภาพของสมาชิกตัวนั้น โดยจะแสดงสมาชิกแต่ละตัวเป็นรูปสี่เหลี่ยมโดย start\_x, start\_y จะเป็นจุดมุมบนซ้าย และ end\_x, end\_y จะกำหนดจุดมุมล่างด้านขวา f\_type จะเป็นตัวกำหนดชนิดของเครื่องมือสำหรับวาดภาพ โดยได้กำหนดชนิดของเครื่องมือวาดภาพเป็นแม่โครไว้ในแฟ้มข้อมูล "menuhead.hpp" ดังนี้

- 1) LINE เครื่องมือสำหรับวาดเส้นตรง
- 2) BRUSH เครื่องมือสำหรับวาดรูปแบบอิสระ
- 3) RECT เครื่องมือสำหรับวาดรูปสี่เหลี่ยมแบบโปร่ง
- 4) FILLRECT เครื่องมือสำหรับวาดรูปสี่เหลี่ยมแบบทึบ
- 5) CIRCLE เครื่องมือสำหรับวาดรูปวงรี (วงกลม) แบบโปร่ง
- 6) FILLCIRCLE เครื่องมือสำหรับวาดรูปวงรี (วงกลม) แบบทึบ
- 7) ERASER เครื่องมือสำหรับการลบภาพแบบอิสระ
- 8) CURVE เครื่องมือสำหรับวาดภาพเส้นโค้ง
- 9) TEXT เครื่องมือสำหรับป้อนข้อความ
- 10) ROLLER เครื่องมือสำหรับระบายสี
- 11) SPRAY\_CAN เครื่องมือสำหรับการพ่นสี
- 12) SELECTION เครื่องมือสำหรับการเลือกบริเวณในจอภาพ

สำหรับ t\_clr และ bg\_clr bor\_clr จะเป็นสีของเครื่องมือ สีพื้น และสีขอบของสมาชิกแต่ละตัวของรายการเลือกเครื่องมือสำหรับวาดภาพนี้

ตัวอย่างการใช้งาน

```
iconmenu_class icon_obj;
icon_obj.init( 5 );
icon_obj.create_menu( 0, 50, 50, 82, 82, LINE );
icon_obj.create_menu( 1, 84, 50, 116, 82, BRUSH );
...
```



```
int get_item_hit( int x_pos, int y_pos )
```

ใช้สำหรับตรวจสอบว่า มีการเลือกเครื่องมือใดจากรายการเลือกสำหรับเครื่องมือวาดภาพนี้ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของตัวชี้ตำแหน่งบนจอภาพ

ตัวอย่างการใช้งาน

```
iconmenu_class icon_obj;
int i;
...
i = icon_obj.get_item_hit( 50, 50 )
```

```
void draw_menu( void )
```

ใช้สำหรับการแสดงรายการเลือกของขนาดของเส้นบนจอภาพ

ตัวอย่างการใช้งาน

```
iconmenu_class icon_obj;
...
icon_obj.draw_menu();
```

## 25. คลาสสำหรับจัดการไทเทิลบาร์

คลาสนี้เป็นดีโอฟิลคลาสของคลาส gmenu\_item\_class และ gmenu\_class

หน้าที่ : จัดการเกี่ยวกับไทเทิลบาร์

ชื่อคลาส : titlebar\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : menuhead.hpp

วิธีการ :

```
void init_title( int left, int top, int right, int bottom, int mx, int my, char *mes, int fx, int fy,
char *fname = "NONAME.PCX", int p_clr = MAGENTA, int b_clr = BLACK,
int t_clr = WHITE )
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับไทเทิลบาร์ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้แสดงได้ดังนี่คือ left, top, right, bottom จะเป็นจุดขอบของบริเวณที่จะแสดงไทเทิลบาร์บนจอภาพ โดย left, top จะเป็นจุดมุมบนด้านซ้าย ส่วน right, bottom จะเป็นจุดมุมล่างด้านขวา (กำหนดเป็นพิกเซล) mx, my ตำแหน่งเริ่มต้นของข้อความ (กำหนดเป็นคอลลัมน์ และแถว โดยคำนวณจากความกว้าง และความสูงของข้อมูลตัวอักษรที่ใช้) fx, fy ตำแหน่งเริ่มต้นของชื่อเพิ่มข้อมูล (กำหนดโดยวิธีเดียวกับตำแหน่งเริ่มต้นของข้อความ) สำหรับ p\_clr, b\_clr และ t\_clr จะเป็นสีพื้นสีของขอบ และสีของตัวอักษรของไทเทิลบาร์ตามลำดับ

ตัวอย่างการใช้งาน

```
titlebar_class titlebar_obj;
titlebar_obj.init_title( 0, 0, 640, 20, 2, 1, "CUPAINT", 10, 1 );
```

```
void draw_bar( void )
```

ใช้สำหรับการแสดงไทเทิลบาร์บนจอภาพ

ตัวอย่างการใช้งาน

```
titlebar_class titlebar_obj;
titlebar_obj.init_title( 0, 0, 640, 20, 2, 1, "CUPAINT", 10, 1 );
titlebar_obj.draw_bar();
```

## 26. คลาสสำหรับการจัดการพื้นที่ที่ใช้วาดภาพ

คลาสนี้เป็นดีโพรไฟคลาสของคลาส gmenu\_class

หน้าที่ : จัดการเกี่ยวกับพื้นที่ที่ใช้ในการวาดภาพ

ชื่อคลาส : drawing\_area\_class

เพิ่มข้อมูลที่กำหนดนิยามของคลาส.: menuhead.hpp

วิธีการ :

```
void init_area( int left, int top, int right, int bottom, int bor_clr = BLACK, int p_clr = WHITE )
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับพื้นที่ที่ใช้ในการวาดภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่จะกำหนดให้เป็นบริเวณที่ใช้วาดภาพ โดย left, top จะหมายถึงจุดมุมบนด้านซ้าย และ right, bottom จะหมายถึงจุดมุมล่างด้านขวา ส่วน bor\_clr และ p\_clr จะเป็นสีของขอบ และสีพื้นของบริเวณที่ใช้ในการวาดภาพ ตามลำดับ

ตัวอย่างการใช้งาน

```
drawing_area_class draw_area_obj;
draw_area_obj.init_area( 87, 40, 587, 404 );
```

```
void draw_area( void )
```

ใช้สำหรับการแสดงพื้นที่ที่ใช้ในการวาดภาพบนจอภาพ

ตัวอย่างการใช้งาน □

```
drawing_area_class draw_area_obj;
draw_area_obj.init_area( 87, 40, 587, 404 );
draw_area_obj.draw_area();
```

## 27. คลาสสำหรับการจัดการรูปภาพ

หน้าที่ : จัดการเกี่ยวกับรูปภาพ

ชื่อคลาส : pictureclass

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : picture.hpp

วิธีการ :

```
void init_picture( int x1, int y1, int x2, int y2 )
```

ใช้สำหรับกำหนดข้อมูลที่จำเป็นสำหรับข้อมูลรูปภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ บริเวณสำหรับการวาดรูปภาพ โดย x1, y1 จะเป็นมุมบนด้านซ้าย และ x2, y2 จะเป็นมุมล่างด้านขวา ของบริเวณที่ใช้วาดภาพ

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.init_picture( 87, 40, 504, 404 );
```

```
void clear_largepic( void )
```

ใช้สำหรับคืนหน่วยความจำที่ใช้ในการเก็บรูปภาพ

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.clear_largepic();
```

```
void updatelargepic( int x, int y, int x1, int y1 )
```

ใช้สำหรับเขียนข้อมูลจากหน่วยความจำของจอภาพลงในหน่วยความจำ ที่ใช้เก็บรูปภาพอาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ บริเวณของรูปภาพบนจอภาพที่ต้องการจะเขียนลงหน่วยความจำ โดยกำหนดเป็นมุมบนด้านซ้าย และมุมล่างด้านขวา

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.updatelargepic( 87, 40, 504, 404 );
```

```
void panlargepic( int x, int y, int x1, int y1 )
```

ใช้สำหรับดูรูปภาพที่ใหญ่เกินกว่าหนึ่งจอภาพของจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ บริเวณของจอภาพที่จัดไว้สำหรับวาดภาพ โดยกำหนดเป็นมุมบนด้านซ้าย และมุมล่างด้านขวา

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.panlargepic( 87, 40, 504, 404 );
```

```
int rotate90( int x1, int y1, int color )
```

ใช้สำหรับหมุนรูปภาพทั้งรูปไปเป็นมุม 90 องศา อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดมุมด้านซ้ายของพื้นที่ที่ใช้วาดภาพ และสีที่กำลังใช้ในการวาดภาพ

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.rotate90( 87, 40, BLACK );
```

```
int rotatecutlarge90( int x1, int y1, int x2, int y2, int right, int bottom, int color )
```

ใช้สำหรับหมุนรูปภาพในบริเวณที่เลือกไว้ไปเป็นมุม 90 องศา อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่ต้องการหมุนรูปภาพ โดยกำหนดเป็นจุดมุมบนด้านซ้าย และมุมล่างด้านขวา ตามลำดับ อาร์กิวเมนต์ right และ bottom จะหมายถึงตำแหน่งขวาสุด และตำแหน่งล่างสุดของบริเวณที่กำหนดไว้สำหรับการวาดภาพตามลำดับ โดยอาร์กิวเมนต์สองตัวนี้จะมีประโยชน์ในการควบคุม ไม่ให้รูปที่หมุนไปมีขนาดเกินกว่าพื้นที่ที่ใช้ในการวาดภาพ สำหรับอาร์กิวเมนต์ color คือสีที่กำลังใช้ในการวาดภาพ

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.rotatecutlarge90( 90, 100, 140, 150, 504, 404, BLACK );
```

```
int rotate180( int x1, int y1 )
```

ใช้สำหรับหมุนรูปภาพทั้งรูปไปเป็นมุม 180 องศา อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดมุมด้านซ้ายของพื้นที่ที่ใช้วาดภาพ (กำหนดเป็นพิกเซล)

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.rotate180( 87, 40 );
```

ศูนย์จักษุวิทยา  
จุฬาลงกรณ์มหาวิทยาลัย

```
int rotatecutlarge180( int x1, int y1, int x2, int y2, int right, int bottom )
```

ใช้สำหรับหมุนรูปภาพในบริเวณที่เลือกไว้ไปเป็นมุม 180 องศา อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่ต้องการหมุนรูปภาพ โดยกำหนดเป็นจุดมุมบนด้านซ้าย และมุมล่างด้านขวาตามลำดับ อาร์กิวเมนต์ right และ bottom จะหมายถึงตำแหน่งขวาสุด และตำแหน่งล่างสุดของบริเวณที่กำหนดไว้สำหรับการวาดภาพตามลำดับ โดยอาร์กิวเมนต์สองตัวนี้จะมีประโยชน์ในการควบคุมไม่ให้รูปที่หมุนไปมีขนาดเกินกว่าพื้นที่ที่ใช้ในการวาดภาพ

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.rotatecutlarge180( 90, 100, 140, 150, 504, 404, BLACK );
```

```
void zoompicture( int x1, int y1, int x2, int y2, int start_x, int start_y, int xspan )
```

ใช้สำหรับการขยายรูปภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่ต้องการขยายรูปภาพ โดยกำหนดเป็นจุดมุมบนด้านซ้าย และมุมล่างด้านขวาตามลำดับ start\_x และ start\_y จะเป็นตำแหน่งเริ่มต้นที่จะให้แสดงรูปภาพที่ขยายแล้ว xspan จะเป็นตัวบอกว่าจะขยายรูปภาพขึ้นมากี่เท่า

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.zoompicture( 50, 50, 100, 100, 87, 40, 4 );
```

```
char far* get_address_monopic( void )
```

ใช้สำหรับรายงานตำแหน่งของหน่วยความจำที่ใช้ในการเก็บรูปภาพของจอโมโนโครม

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
char far *address;
address = picture_obj.get_address_monopic();
```

```
void get_address_pic( char far *p1, char far *p2, char far *p3, char far *p4 )
```

ใช้สำหรับรายงานตำแหน่งของหน่วยความจำที่ใช้ในการเก็บรูปภาพของจอ VGA/EGA อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตัวแปรที่จะใช้เก็บค่าหน่วยความจำของรูปภาพสำหรับแต่ละเพลน(Plane)

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
char far *address1, *address2, *address3, *address4;
picture_obj.get_address_pic( address1, address2, address3, address4 );
```

```
int save_pic( int left, int top, int right, int bottom, int off_point[], int *count )
```

ใช้สำหรับเก็บรูปภาพจากบริเวณที่ผู้ใช้เลือกลงคลิปบอร์ด อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่ต้องการจะจัดเก็บลงคลิปบอร์ด โดยกำหนดเป็นจุดมุมบนด้านซ้าย และมุมล่างด้านขวา ตามลำดับ สำหรับ off\_point จะเก็บตำแหน่งของจุดในตำแหน่งที่มีการแบ่งรูปภาพ (ที่ต้องแบ่งรูปภาพเนื่องจากวิธีการนี้จะเรียกใช้ฟังก์ชันการจัดเก็บรูปภาพลงหน่วยความจำ ของตัวแปลโปรแกรม TURBO C++ ซึ่งจะเก็บรูปภาพที่มีขนาดใหญ่เกิน 64 กิโลไบต์ไม่ได้) ส่วน count จะเก็บจำนวนครั้งของการแบ่งรูปภาพ

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
int off_point[20], count = 0;
picture_obj.save_pic( 90, 200, 160, 300, off_point, &count );
```

```
void writepic( int left, int top, int write_mode, void far *buff[], int off_point[], int count )
```

ใช้สำหรับเขียนข้อมูลจากคลิปบอร์ดลงบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งบนจอภาพที่ต้องการจะเขียนข้อมูลจากคลิปบอร์ด วิธีการที่จะใช้เขียนรูปภาพ ตำแหน่งของหน่วยความจำที่ใช้เป็นคลิปบอร์ด ส่วน off\_point และ count ขอให้ดูจาก savepic()

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.writepic( left, top, COPY_PUT, buff, off_point, count );
```

```
void freepic( void far *buff[], int *count )
```

ใช้สำหรับการคืนหน่วยความจำให้กับระบบเมื่อไม่ต้องการใช้คลิปบอร์ด อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ตำแหน่งของหน่วยความจำที่ใช้เป็นคลิปบอร์ด และจำนวนครั้งที่มีการแบ่งรูปภาพในขณะที่จัดเก็บข้อมูลลงคลิปบอร์ด (ดู save\_pic())

ตัวอย่างการใช้งาน

```
pictureclass picture_obj;
picture_obj.freepic( buff, count );
```

## 28. คลาสสำหรับการจัดการรูปภาพ PCX และ TIFF

คลาสนี้เป็นดีไรฟ์คลาสของคลาส pictureclass

หน้าที่ : จัดการเกี่ยวกับรูปภาพภาพ PCX และ TIFF

ชื่อคลาส : pcxandtiffclass

เพิ่มข้อมูลที่กำหนดนิยามของคลาส : picture.hpp

วิธีการ :

```
int initpic(char *fname )
```

ใช้สำหรับกำหนดค่าที่จำเป็นสำหรับการอ่าน หรือเขียนรูปภาพ PCX หรือ TIFF จาก (ลง) เพิ่มข้อมูล อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ ชื่อของเพิ่มข้อมูลภาพ

ตัวอย่างการใช้งาน

```
pcxandtiffclass pic_obj;
pic_obj.initpic( "TEST.PCX" )
```

```
void savepic(int x1, int y1, int x2, int y2)
```

ใช้สำหรับเขียนข้อมูลรูปภาพจากบริเวณที่ผู้ใช้กำหนด ลงเพิ่มข้อมูลโดยจะจัดเก็บรูปภาพด้วยโครงสร้างเพิ่มข้อมูลกราฟิกแบบ PCX อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่ต้องการจัดเก็บรูปภาพ โดยกำหนดเป็นจุดมุมบนด้านซ้าย และมุมล่างด้านขวาตามลำดับ

ตัวอย่างการใช้งาน

```
pcxandtiffclass pic_obj;
...
pic_obj.savepic( 90, 200, 150, 300 );
```

```
void savelargepic( void )
```

ใช้สำหรับเขียนข้อมูลรูปภาพทั้งรูปลงเพิ่มข้อมูล โดยจะจัดเก็บรูปภาพด้วยโครงสร้างเพิ่มข้อมูลกราฟิกแบบ PCX

ตัวอย่างการใช้งาน

```
pcxandtiffclass pic_obj;
...
pic_obj.savepic();
```

```
int loadsmallpic(int x1, int y1, int x2, int y2)
```

ใช้สำหรับอ่านข้อมูลรูปภาพบางส่วนมาแสดงในบริเวณที่ผู้ใช้กำหนดบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่ต้องการแสดงรูปภาพ โดยกำหนดเป็นจุดมุมบนด้านซ้ายและมุมล่างด้านขวา ตามลำดับ

ตัวอย่างการใช้งาน

```
pcxandtiffclass pic_obj;
...
pic_obj.loadsmallpic( 90, 200, 150, 300 );
```

```
int loadpic(int x1, int y1, int x2, int y2)
```

ใช้สำหรับอ่านข้อมูลรูปภาพทั้งรูปภาพมาแสดงบนจอภาพ อาร์กิวเมนต์ที่ต้องส่งให้กับวิธีการนี้คือ จุดขอบของบริเวณที่ใช้ในการวาดภาพภาพโดยกำหนดเป็นจุดมุมบนด้านซ้าย และมุมล่างด้านขวา ตามลำดับ

ตัวอย่างการใช้งาน

```
pcxandtiffclass pic_obj;
...
pic_obj.loadpic( 90, 200, 150, 300 );
```



ประวัติผู้เขียน

ชื่อ นายศรัณย์ อินทโกสุม เกิดวันที่ 27 กรกฎาคม พ.ศ. 2510 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2531 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อพ.ศ. 2532



ศุภชัยวิทย์ทรัพย์ากร  
จุฬาลงกรณ์มหาวิทยาลัย