บทที่ 4

ผลการศึกษาฟิกส์และโปรแกรมกรณีศึกษา

การทำวิจัยนี้มีจุดประสงค์อยู่ที่การศึกษาความสามารถและการใช้งานฟิกส์โดยใช้ซันฟิกส์รุ่น 2.0 ปัญหาและแนวทางการแก้ปัญหาเมื่อนำซันฟิกส์รุ่น 2.0 มาใช้ในการพัฒนาโปรแกรมด้านกราฟิก และแสดงตัวอย่างของโปรแกรมที่ถูกพัฒนาด้วยซันฟิกส์รุ่น 2.0

ในบทนี้จะกล่าวถึงผลของการศึกษาฟิกส์ และผลของการพัฒนาโปรแกรมกรณีศึกษาโดยใช้ซันฟิกส์รุ่น 2.0 ส่วนแนวทางการแก้ไขปัญหาในการพัฒนาโปรแกรมบางประการของซันฟิกส์รุ่น 2.0 ผู้วิจัยได้กล่าวไว้แล้วในบทที่ 3 การออกแบบและพัฒนาโปรแกรม หัวข้อการทำงานของโปรแกรม

ผลการศึกษาฟิกส์

จากการศึกษาและทดลองใช้ซันฟิกส์รุ่น 2.0 ในการพัฒนาโปรแกรมสร้างแบบจำลอง 3 มิติ ทำให้พบว่าฟิกส์และซันฟิกส์รุ่น 2.0 มีทั้งจุดเด่นและจุดด้อยดังนี้

1. จุดเด่นของฟิกส์

จุดเด่นของฟิกส์จะเป็นเรื่องของความง่ายในการใช้งาน และคุณสมบัติใช้ได้กับหลายระบบ ส่วนซันฟิกส์รุ่น 2.0 นั้นสามารถรักษาความเป็นมาตรฐานของฟิกส์ไว้ได้ดี ดังนั้นซันฟิกส์จึงไม่มีจุดเด่นพิเศษที่ทำให้แตกต่างจากการทำให้เกิดผลของฟิกส์ตัวอื่นๆ

1.1 ความง่ายในการใช้งาน

เมื่อเข้าใจแนวคิดหลักและลักษณะการทำงานของฟิกส์แล้ว การใช้ฟิกส์เพื่อพัฒนาโปรแกรมจะทำได้ค่อนข้างง่าย ขั้นตอนการทำงานของฟิกส์ไม่ซับซ้อน ไม่มีการกระโดดไปมา การทำงานเป็นแบบเรียงลำดับ และฟังก์ชันประเภทรับข้อมูลและแสดงผลมักจบการทำงานภายในฟังก์ชันเดียว ไม่ว่าจะเป็นฟังก์ชันกราฟิกง่ายๆ เช่น การสร้างโพลีไลน์ หรือฟังก์ชันกราฟิกขั้นสูง เช่น การสร้างเส้นโค้งบีสไปลน์ ฟิกส์ช่วยให้นักพัฒนาโปรแกรมสามารถสร้างโปรแกรมกราฟิกที่ดีได้ โดยไม่จำเป็นต้องเรียนรู้ทฤษฎีทางด้านกราฟิกมากนัก เช่น การให้แสงกับวัตถุสามารถทำได้โดยการกำหนดตำแหน่งและคุณสมบัติของแหล่งกำเนิดแสง รวมทั้งกำหนดลักษณะการสะท้อนแสงของวัตถุได้ โดยไม่ต้องเข้าใจถึงทฤษฎีของแสง การสร้างเส้นโค้งบีสไปลน์

สามารถทำได้โดยไม่ต้องเรียนรู้ทฤษฎีของเส้นโค้งบีสไปลน์อย่างลึกซึ้ง อย่างไรก็ตามผู้ที่เข้าใจทฤษฎีด้านคอมพิวเตอร์กราฟิกเป็นอย่างดี ก็จะทำให้งานที่พัฒนาขึ้นจากฟิกส์นั้นมีคุณภาพและตรงตามความต้องการมากขึ้น

1.2 คุณสมบัติการไม่ขึ้นกับอุปกรณ์จริงใดๆ และคุณสมบัติใช้ได้กับหลายระบบ

อุปกรณ์ต่างๆ ของฟิกส์ที่ใช้ในการเขียนโปรแกรมจะเป็นอุปกรณ์เชิงตรรกะทั้งสิ้น ไม่ว่าจะเป็นอุปกรณ์นำเข้าหรืออุปกรณ์ส่งออกก็ตาม ทำให้นักพัฒนาโปรแกรมไม่ต้องเขียนโปรแกรมเฉพาะเพื่อจัดการกับอุปกรณ์จริงที่ต่างกันอย่างเช่นที่เกิดกับโปรแกรมบนเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วๆ ไป การใช้อุปกรณ์เชิงตรรกะจะทำให้ผู้ใช้โปรแกรมสามารถเปลี่ยนแปลงอุปกรณ์จริงได้ตลอดเวลา

โปรแกรมที่ถูกพัฒนาขึ้นด้วยฟิกส์อย่างเดียวนั้น จะมีคุณสมบัติใช้ได้กับหลายระบบและสามารถทำงานในระบบเครือข่ายได้ ถ้าเครื่องที่ใช้สนับสนุนฟิกส์และเพกซ์ นอกจากการใช้ข้ามอุปกรณ์แล้วฟิกส์ยังสามารถใช้งานข้ามระบบได้ด้วย เช่น สามารถทำงานได้กับทั้งเอกซ์วินโดว์ที่มีตัวประสานกับผู้ใช้แบบกราฟิกชนิดโอเพนลุกและโมทิฟ

สำหรับการทำให้เกิดผลของฟิกส์ที่รักษามาตรฐานของฟิกส์ไว้ได้ โปรแกรมที่ถูกพัฒนาโดยการทำให้เกิดผลของฟิกส์ตัวนั้นจะสามารถนำไปใช้กับการทำให้เกิดผลของฟิกส์ตัวอื่นได้ด้วย โดยการแก้ไขรหัสคำสั่ง (Code) บางส่วนเท่านั้น เช่น การนำโปรแกรมฟิกส์ง่ายๆ ของเดก ฟิกส์ (DEC PHIGS) มาใช้กับซันฟิกส์รุ่น 2.0 สามารถทำได้โดยการแปลงชนิดเวิร์กสเตชันของเดก ฟิกส์ มาเป็นเวิร์กสเตชันชนิดที่ซันฟิกส์รุ่น 2.0 รู้จัก เป็นต้น

2. <u>จุดด้อยของฟิกส์</u>

จุดด้อยของฟิกส์ คือ เรื่องของการเตรียมการจัดการกับข้อมูลในหน่วยเก็บโครงสร้างส่วนกลางได้ไม่ดีพอ เนื่องจากฟิกส์เป็นระบบกราฟิกที่ใช้ฐานข้อมูลกราฟิก ฟิกส์จึงควรเตรียมการจัดการกับฐานข้อมูลนี้ให้มีประสิทธิภาพมากขึ้น

2.1 การค้นหารูปทรงพื้นฐานทำได้วิธีเดียว

ฟิกส์เตรียมการค้นหารูปทรงพื้นฐานในหน่วยเก็บโครงสร้างส่วนกลางไว้เพียงวิธีเดียวคือ การค้นด้วยข้อมูลเรขาคณิต การค้นหารูปทรงพื้นฐานที่ตั้งอยู่ในขอบเขตทางเรขาคณิตที่กำหนดสามารถทำได้ แต่จะไม่สามารถค้นหารูปทรงพื้นฐานจากข้อมูลอื่นๆ ได้ เช่น ไม่สามารถค้นหารูปทรงพื้นฐานเฉพาะที่มีสีแดง หรือรูปทรงพื้นฐานเฉพาะที่มีการสะท้อนแสงแบบล้อมรอบ (Ambient) ได้

2.2 ข้อมูลที่ส่งกลับมาจากการค้นหายังไม่ละเอียดพอ

เมื่อค้นหารูปทรงพื้นฐานซึ่งอยู่ในขอบเขตที่ค้นหาพบ ฟิกส์จะส่งข้อมูลของการค้นพบกลับมาเป็นตัวระบุโครงสร้างและตำแหน่งขององค์ประกอบของรูปทรงพื้นฐานในโครงสร้างนั้น ซึ่งเป็นข้อมูลที่ละเอียดที่สุดที่ฟิกส์จะส่งกลับมา ฟิกส์จะไม่บอกว่าข้อมูลใดของรูปทรงพื้นฐานที่ทำให้รูปทรงพื้นฐานนั้นถูกค้นพบ เช่น จุดยอดใดบ้างของโพลีไลน์ที่อยู่ในขอบเขตของการค้นหา หรือเมื่อผู้ใช้ใช้อุปกรณ์นำเข้าแบบพิกเลือกโพลีไลน์เข้ามา จะไม่สามารถรู้ได้ว่าผู้ใช้ได้เลือกจุดยอดหรือส่วนของเส้นตรงใดเข้ามา ทำให้การตัดสินใจว่าผู้ใช้ต้องการเลือกแก้ไขอะไรในรูปทรงพื้นฐานนั้นทำได้ลำบากขึ้น

2.3 การแก้ไขข้อมูลขององค์ประกอบในโครงสร้างค่อนข้างยุ่งยาก

การแก้ไของค์ประกอบในโครงสร้างสามารถทำได้ 3 วิธี คือ การแทรก การลบ (DELETE) และการแทนที่ แต่จะไม่สามารถจัดการกับข้อมูลภายในขององค์ประกอบนั้นได้โดยตรง เช่น ไม่สามารถเพิ่มจุดยอดในองค์ประกอบที่เป็นโพลีไลน์ได้ หรือไม่สามารถแก้ไขข้อมูลสีในองค์ประกอบที่เป็นลักษณะประจำสีได้ เป็นต้น การแก้ไขข้อมูลเหล่านี้สามารถทำได้โดยการดึงข้อมูลขององค์ประกอบนั้นมาจากหน่วยเก็บโครงสร้างส่วนกลาง แล้วทำการแก้ไข จากนั้นจึงแทนที่องค์ประกอบนั้นด้วยข้อมูลใหม่ การดึงข้อมูลจากหน่วยเก็บโครงสร้างส่วนกลางออกมาเป็นเรื่องยุ่งยากต้องทำหลายขั้นตอน และทำให้โปรแกรมทำงานผิดพลาดได้บ่อย ดังนั้นการแก้ไขข้อมูลเพียงบางส่วนหรือข้อมูลจำนวนมากก็ยุ่งยากไม่แตกต่างกัน ในกรณีของการแก้ไขข้อมูลทั้งหมดขององค์ประกอบนั้น จะทำได้ง่ายกว่าการแก้ไขข้อมูลเพียงบางส่วน เพราะไม่ต้องดึงข้อมูลเก่าออกมาจากหน่วยเก็บโครงสร้างส่วนกลาง

2.4 การเปรียบเทียบข้อมูลและความซ้ำซ้อนของการเก็บข้อมูลหน่วยเก็บโครงสร้างส่วนกลาง

ฟิกส์มิได้จัดเตรียมการเปรียบเทียบข้อมูลในหน่วยเก็บโครงสร้างส่วนกลางไว้ให้ โดยเฉพาะในกรณีของโครงสร้าง เมื่อฟิกส์ไม่สามารถเปรียบเทียบโครงสร้างได้ ดังนั้นโครงสร้างที่มีข้อมูลเหมือนกันแต่มีตัวระบุโครงสร้างต่างกัน จะถูกเก็บไว้ในหน่วยเก็บโครงสร้างส่วนกลางทั้งหมดทุกโครงสร้าง ซึ่งจะทำให้หน่วยเก็บโครงสร้างส่วนกลางมีขนาดใหญ่ขึ้นโดยไม่จำเป็น

2.5 การใช้ตัวระบุเป็นจำนวนเต็ม (Integer)

ฟิกส์จะใช้ตัวระบุ เพื่อใช้ในการอ้างอิงสิ่งต่างๆ ของฟิกส์ที่สามารถทำงานได้พร้อมกันหลายๆ ตัว เช่น เวิร์กสเตชัน โครงสร้าง เป็นต้น ข้อดีของการใช้ตัวระบุเป็นจำนวนเต็มคือ จะทำให้ง่ายต่อการใช้งาน แต่จะมีปัญหาเรื่องการจัดการไม่ให้ตัวระบุซ้ำกัน โดยเฉพาะกรณีของตัวระบุโครงสร้าง เช่น เมื่อสร้างโครงสร้างขึ้นมา 1 โครงสร้าง โดยกำหนดให้มีตัวระบุโครง

สร้างเป็น 10 แล้วเก็บโครงสร้างนี้ลงในแฟ้มเก็บถาวร ตัวระบุโครงสร้างในแฟ้มเก็บถาวรก็จะเป็น 10 ด้วย เมื่อลบโครงสร้างนี้ออกจากโปรแกรม ค่าตัวระบุโครงสร้าง 10 ก็จะว่าง โปรแกรมจะสามารถสร้างโครงสร้างใหม่โดยใช้ตัวระบุโครงสร้าง 10 ได้อีก ถ้าโปรแกรมได้สร้างโครงสร้างใหม่โดยใช้ตัวระบุโครงสร้าง 10 เมื่อโปรแกรมไปดึงเอาโครงสร้างในแฟ้มเก็บถาวรเดิมนี้ขึ้นมา ก็จะมีปัญหาตัวระบุโครงสร้างซ้ำกันขึ้น

ฟิกส์ควรจะใช้ตัวระบุเป็นข้อมูลชนิดตัวชี้หรือค่าควบคุมแบบในเอกซ์วิวมากกว่าที่จะใช้ตัวระบุเป็นค่าจำนวนเต็มโดดๆ

3. จุดด้อยของซันฟิกส์รุ่น 2.0

จุดด้อยของซันฟิกส์รุ่น 2.0 มี 2 เรื่องใหญ่ๆ คือ โครงสร้างข้อมูล และการทำงาน

3.1 โครงสร้างข้อมูลบางตัวซ้ำซ้อนกัน

ซันฟิกส์รุ่น 2.0 มีโครงสร้างข้อมูลที่มีรูปแบบต่างกัน แต่ใช้ทำงานได้เหมือนกัน ซึ่งเป็นการซ้ำซ้อนโดยไม่จำเป็น เช่น โครงสร้างข้อมูลของการกำหนดข้อมูลสีโดยตรง ได้แก่ Pgcolr ใช้สำหรับกำหนดลักษณะประจำสีให้รูปทรงพื้นฐาน และ Pcoval ใช้สำหรับกำหนดสีที่จุดยอดของรูปทรงพื้นฐาน โดยมีรูปแบบดังนี้

```
typedef  struct {
   Pint  type;                  /* indirect, RGB, CIE, HSV, HLS */
   union  {
     Pint  ind;                 /* color table index */
     struct  {
       Pfloat  x;               /* red, hue, cieluv_x */
       Pfloat  y;               /* green, saturation, lightness, cieluv_y */
       Pfloat  z;               /* blue, value, saturation, cieluv_y_lum */
     } general;
   } val;
}Pgcolr;
typedef  union  {
   Pint  ind;                   /* color index */
   Pcolr_rep  direct;           /* direct color */
} Pcoval;
typedef  union  {
```

```
    Prgb  rgb;              /* RED, GREEN, BLUE */
    Pcieluv  cieluv;        /* CIELUV */
    Phls  hls;              /* Hue lightness Saturation */
    Phsv  hsv;              /* Hue saturation value */
    Pdata  unsupp;          /* unsupport color model */
  } Pcolr_rep;
```

ทั้ง Pcoval และ Pgcolr สามารถใช้กำหนดข้อมูลสีได้เหมือนกัน ซันฟิกส์รุ่น 2.0 ควรเลือกใช้เพียงอย่างใดอย่างหนึ่ง

3.2 โครงสร้างข้อมูลบางตัวซับซ้อนและยาวเกินไปโดยไม่จำเป็น

ซันฟิกส์รุ่น 2.0 จะรวบรวมข้อมูลที่เกี่ยวข้องกันมาไว้ในโครงสร้างเดียวกัน โครงสร้างข้อมูลบางตัวมีข้อมูลที่เกี่ยวข้องกันจำนวนมาก การใช้โครงสร้างข้อมูลตัวนั้นจึงค่อนข้างยุ่งยาก ทำให้เกิดความสับสนได้ง่าย และยังทำให้รหัสคำสั่งของโปรแกรมยาวโดยไม่จำเป็นอีกด้วย เช่น การอ้างถึงข้อมูลของจุด x ของ โครงสร้างข้อมูลของฟังก์ชันหลีก (Escape Function) หมายเลข 5 จะต้องผ่านโครงสร้างข้อมูลถึง 4 โครงสร้างข้อมูล ดังนี้

```
esc_in.escape_in_u5.points.points[0].x
```

3.3 โครงสร้างข้อมูลของฟิกส์และฟิกส์พลัสไม่สามารถใช้ด้วยกันได้

โครงสร้างข้อมูลของฟิกส์และฟิกส์พลัสที่เกี่ยวข้องกัน ไม่สามารถโอนใช้กันได้โดยตรง ต้องมีการแปลงจากโครงสร้างข้อมูลชนิดหนึ่งไปเป็นอีกชนิดหนึ่งก่อน จึงจะสามารถใช้งานได้ เช่น กรณีของโพลีไลน์กับกลุ่มของโพลีไลน์ที่กำหนดด้วยข้อมูล ตัวอย่างโพลีไลน์ 3 มิติ จะมีโครงสร้างข้อมูลดังนี้

```
typedef  struct  {
  Pint  num_points;         /* จำนวนของจุดยอดบนโพลีไลน์ */
  Ppoint3  *point;          /* array ของจุด */
} Ppoint_list3;
```

ส่วนกลุ่มของโพลีไลน์ที่กำหนดด้วยข้อมูลมีโครงสร้างข้อมูลดังนี้

```
typedef  struct  {
  Pint  num_vertices;               /* จำนวนของ vertex */
  Pline_vdata_arr3  vertex_data     /* vertex data */
} Pline_vdata_list3;
typedef  union  {
```

```
    Ppoint3  *points;          /* array of points กรณีที่ไม่กำหนดข้อมูลสีที่ vertex*/
    Pptco3  *ptcolrs;          /* array of points with colr กรณีที่กำหนดข้อมูลสีที่ vertex */
  } Pline_vdata_arr3;
  typedef  struct  {
    Ppoint3  point;            /* point */
    Pcoval  colr;              /* color data */
  } Pptco3;
```

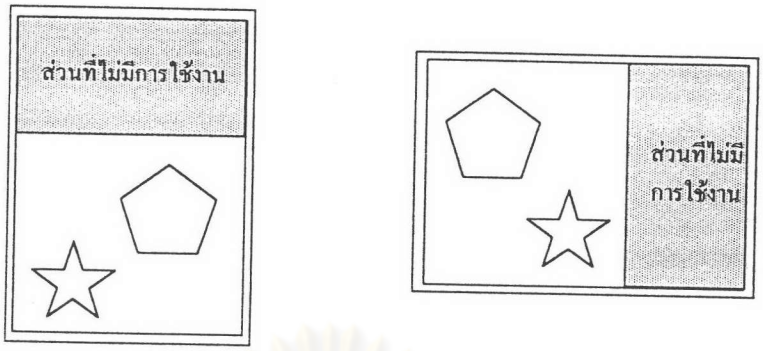
ถ้าต้องการนำข้อมูลจุดของกลุ่มของโพลีไลน์ที่กำหนดด้วยข้อมูลแบบที่กำหนดข้อมูลสีที่จุดยอดมาใช้กับโพลีไลน์จะต้องแปลงจาก Pptco3 มาเป็น Ppoint3 ก่อนจึงจะใช้ได้ แต่ถ้าแก้โครงสร้างข้อมูลของกลุ่มของโพลีไลน์ที่กำหนดด้วยข้อมูลเป็น

```
  typedef  struct  {
    Pint  num_vertices;        /* จำนวนของ vertex */
    Ppoint3  *points;          /* array ของข้อมูลจุด */
    Pcoval  *colr;             /* array ของข้อมูลสี */
  } Pline_vdata_list3;
```

โครงสร้างข้อมูลใหม่นี้จะสามารถดึงข้อมูลจุดไปใช้กับโพลีไลน์ได้ทันที และทำให้โครงสร้างข้อมูลของกลุ่มของโพลีไลน์ที่กำหนดด้วยข้อมูลลดความซับซ้อนลงได้

3.4 การทำงานร่วมกับเอกซ์วินโดว์ยังไม่ดีเท่าที่ควร

ซันฟิกส์รุ่น 2.0 ยังจัดการกับอีเวนต์บางอย่างของเอกซ์วินโดว์ได้ไม่ดีนัก เช่น การเปลี่ยนขนาด (Resize) เป็นต้น เนื่องจากซันฟิกส์รุ่น 2.0 จะรักษารูปร่างวิวพอร์ต (Viewport) ให้เป็นทรงสี่เหลี่ยมจตุรัสเสมอ ถ้าใช้เวิร์กสเตชันแบบเอกซ์ทูล ส่วนแสดงภาพ (Canvas, Display) ของวินโดว์นั้นจะเป็นสี่เหลี่ยมจตุรัสเมื่อเปิดใช้งานครั้งแรกเสมอ ถ้ามีการเปลี่ยนแปลงขนาดของวินโดว์แล้วทำให้ส่วนแสดงภาพไม่เป็นสี่เหลี่ยมจตุรัส ซันฟิกส์รุ่น 2.0 จะสร้างวิวพอร์ตที่เป็นสี่เหลี่ยมจตุรัสที่มีด้านเท่ากับด้านที่สั้นที่สุดของวินโดว์นั้น ซึ่งจะทำให้เกิดช่องว่างขึ้นดังรูป



รูป 4.1 ลักษณะช่องว่างที่เกิดขึ้นจากการพยายามรักษาวิวพอร์ตให้เป็นสี่เหลี่ยมจตุรัส

โดยตัวของซันฟิกส์รุ่น 2.0 เองแล้วจะไม่มีเส้นที่ใช้บอกอาณาเขตของวิวพอร์ต ดังนั้นถ้าโปรแกรมประยุกต์จัดการเรื่องนี้ไม่ดี ผู้ใช้จะไม่มีทางรู้ได้เลยว่าออกนอกขอบเขตของวิวพอร์ตหรือไม่ ส่วนการใช้เวิร์กสเตชันแบบเอกซ์ ดรอเอเบิล ซันฟิกส์รุ่น 2.0 ได้เตรียมฟังก์ชันหลีกหมายเลข 7 ไว้เพื่อใช้ในการเปลี่ยนขนาด ซึ่งการทำงานของฟังก์ชันให้ผลที่ไม่น่าพอใจนัก เพราะทำงานในลักษณะเดียวกับการเปลี่ยนขนาดในเอกซ์ทูล และบางครั้งอาจมีการผิดพลาดด้วย ในการใช้งานจริงจำเป็นต้องใช้ฟังก์ชันนี้ร่วมกับวิธีการจัดการการเปลี่ยนขนาดที่คิดขึ้นเองด้วย

3.5 ส่วนอุปกรณ์นำเข้าเชิงตรรกะของฟิกส์ทำได้ไม่ดี

ปัญหาในส่วนอุปกรณ์นำเข้าของซันฟิกส์รุ่น 2.0 สามารถแยกได้เป็น 2 กรณี คือ ปัญหาของกลุ่มอุปกรณ์นำเข้าที่รับข้อมูลกราฟิก คือ โลเคเตอร์ สโตรกและพิก กับกลุ่มที่ไม่ได้รับข้อมูลกราฟิก คือ แวลูเอเตอร์ ชอยซ์และสายอักขระ

กลุ่มของอุปกรณ์นำเข้าที่รับข้อมูลกราฟิก จะมีปัญหาเกี่ยวกับการจัดการกับอีเวนต์ของเอกซ์วินโดว์ที่นอกเหนือจากอีเวนต์ที่เป็นทริกเกอร์และการหยุด เนื่องจากอุปกรณ์นำเข้าของฟิกส์จะสามารถใช้กับเวิร์กสเตชันได้ชนิดเดียว คือ แบบเอกซ์ทูล ซึ่งเป็นแบบที่ไม่สามารถติดต่อกับเอกซ์วินโดว์โดยตรงได้ จึงไม่สามารถจัดการกับอีเวนต์ใดๆ ของเอกซ์วินโดว์ได้เลย เช่น ถ้ากดปุ่มซ้ายของเมาส์แช่สักพักแล้วปล่อย เมื่อเลื่อนเมาส์จะพบว่ามีภาพตัวพร้อมของอุปกรณ์ 2 ตัว โดยตัวหนึ่งจะเลื่อนตามเมาส์ และอีกตัวหนึ่งจะค้างอยู่ที่เดิมที่ได้กดปุ่มเมาส์แช่ไว้ ซันฟิกส์รุ่น 2.0 จะไม่ลบภาพตัวพร้อมที่ค้างอยู่จนกว่าจะมีการกดปุ่มซ้ายใหม่อีกครั้ง

กลุ่มของอุปกรณ์นำเข้าที่ไม่ได้รับข้อมูลกราฟิก จะถูกสร้างไว้บนวินโดว์ซึ่งมีลักษณะเป็นเฟรมแบบผุดขึ้น โดยอุปกรณ์นำเข้า 1 ตัว จะใช้ 1 วินโดว์ ลักษณะของเฟรมแบบผุดขึ้น คือ จะหายไปเมื่อจบการทำงาน เมื่อเกิดทริกเกอร์ขึ้นอุปกรณ์นั้นจะรับข้อมูลแล้วหายไป ถ้าต้องการใส่ข้อมูลใหม่ต้องไปเรียกอุปกรณ์นำเข้านั้นมาอีกครั้ง ดังนั้นการใช้ภาวะการดำเนินการ (Operation Mode) ของอุปกรณ์นำเข้า ไม่ว่าภาวะใดก็จะเหมือนกับการใช้ภาวะการร้องขอ เพราะ

ไม่สามารถรักษาตัวพร้อมของอุปกรณ์ไว้ได้เมื่อไม่มีการร้องขอ แต่จะต่างกับภาวะการร้องขอตรงที่ไม่มีการบล็อก (Block) โปรแกรมเกิดขึ้น

การใช้เฟรมแบบผุดขึ้นกับแวลูเอเตอร์ที่ใช้ตัวพร้อมที่มีลักษณะเป็นแถบเลื่อนอย่างสไลเดอร์ การใช้สไลเดอร์มักใช้เมื่อข้อมูลนำเข้าที่ต้องการนั้นจะต้องปรับค่าไปมา แต่การใช้เฟรมแบบนี้เมื่อมีการปรับค่า 1 ครั้ง (กดปุ่มบนสไลเดอร์เลื่อนไปแล้วปล่อย) จะถือว่าเป็น 1 ทริกเกอร์ ดังนั้นแวลูเอเตอร์จะรับข้อมูลแล้วหายไป ถ้าต้องการปรับข้อมูลอีกต้องเรียกแวลูเอเตอร์นั้นขึ้นมาใหม่ ซึ่งทำให้การทำงานยุ่งยากขึ้น

เอกซ์วินโดว์มักจะป้องกันการเปิดวินโดว์ใหม่ทับวินโดว์เก่า เอกซ์วินโดว์จึงเปิดวินโดว์ใหม่ให้เหลื่อมกับวินโดว์ที่เปิดอยู่ก่อนหน้า โดยดูจากตำแหน่งซ้ายบนของวินโดว์ที่เปิดอยู่ก่อนหน้า ดังนั้นจึงไม่สามารถกำหนดตำแหน่งที่ปรากฏ (Echo Area) ที่แน่นอนของอุปกรณ์นั้นได้ วินโดว์ของอุปกรณ์จึงปรากฏแบบกระโดดไปมาบนจอภาพ และเนื่องจากซันฟิกส์รุ่น 2.0 ได้สร้างอุปกรณ์ 1 ตัว เป็น 1 วินโดว์ จึงไม่สามารถจัดกลุ่มของอุปกรณ์ให้เป็นกลุ่มเดียวกันได้ เช่น การรับข้อมูลของจุด 3 มิติ 1 จุด จะต้องใช้ 3 วินโดว์ เพื่อรับข้อมูล x, y และ z

3.6 เวิร์กสเตชันไม่สามารถใช้อุปกรณ์นำเข้าของฟิกส์ร่วมกับเอกซ์ทูลคิตได้

จุดเด่นของเอกซ์ทูล คือ สามารถใช้คุณสมบัติของฟิกส์ได้เต็มที่ และมีตัวเฝ้าคุมฟิกส์ (PHIGS Monitor) ที่ใช้ควบคุมการติดต่อกับเอกซ์วินโดว์ ทำให้โปรแกรมประยุกต์ที่ใช้เอกซ์ทูลสามารถสร้างได้ง่าย แต่มีข้อเสีย คือ ประสิทธิภาพค่อนข้างต่ำ ส่วนโปรแกรมประยุกต์ที่ใช้เอกซ์ ดรอเอเบิลจะมีประสิทธิภาพค่อนข้างดี แต่จะยากในการพัฒนา เพราะต้องจัดการกับเอกซ์วินโดว์เอง

ถ้าเลือกที่จะใช้เอกซ์ทูลคิตแล้ว ซันฟิกส์รุ่น 2.0 จะอนุญาติให้ใช้เวิร์กสเตชันได้ชนิดเดียว คือ แบบเอกซ์ ดรอเอเบิล ซันฟิกส์รุ่น 2.0 ไม่มีเวิร์กสเตชันที่สามารถใช้อุปกรณ์นำเข้าของฟิกส์ร่วมกับเอกซ์ทูลคิตได้เลย ซึ่งถ้ามีอาจสามารถใช้โลเคเตอร์ของฟิกส์ และใช้พาเนลของเอกซ์วิวรับข้อมูลแบบแวลูเอเตอร์ ชอยซ์และสายอักขระแทนได้

3.7 ฟิกส์ไม่อนุญาติให้ติดต่อกับข้อมูลซึ่งอยู่ในหน่วยเก็บโครงสร้างส่วนกลางโดยตรง การเรียกข้อมูลในหน่วยเก็บโครงสร้างส่วนกลางออกมาใช้จะทำได้โดยใช้ฟังก์ชันการสอบถาม (Inquiry Function) ซึ่งฟังก์ชันการสอบถามเหล่านี้จะสำเนาข้อมูลจากหน่วยเก็บโครงสร้างส่วนกลางขึ้นมาไว้บนหน่วยความจำ (Memory) หน่วยความจำส่วนนี้จะเรียกว่า สโตร์ (Store) ซันฟิกส์รุ่น 2.0 อนุญาติให้มีการสร้างสโตร์ได้พร้อมกันมากกว่า 1 สโตร์ แต่สโตร์มีขนาดค่อนข้างใหญ่ จึงทำให้สิ้นเปลืองหน่วยความจำมาก และถ้าหน่วยความจำของเครื่องมีจำกัดการใช้สโตร์มัก

ทำให้โปรแกรมมีปัญหาได้ เช่น ในการทำวิจัยนี้ปัญหาที่พบ คือ โปรแกรมจะเกิดผิดพลาด และเกิดการหยุดงานผิดปรกติ แต่บางครั้งอาจทำให้เครื่องหยุด ทำงานต่อไม่ได้เช่นกัน

สโตร์ 1 ตัวจะใช้ได้เพียง 1 ครั้ง การใช้สโตร์ซ้ำกันจะทำให้โปรแกรมเกิดการหยุดงานผิดปรกติได้ ดังนั้นเมื่อจะใช้ฟังก์ชันการสอบถามที่ต้องการใช้สโตร์ จะต้องสร้างสโตร์ใหม่เสมอ ในตัวโปรแกรมจะต้องมีการลบและสร้างสโตร์ใหม่เสมอ บางครั้งอาจทำให้การทำงานของโปรแกรมช้าลง และถ้าเป็นโปรแกรมประเภทเชิงโต้ตอบ (Interactive) อาจทำให้การทำงานผิดพลาดได้ เช่น การใช้เวิร์กสเตชันแบบเอกซ์ ดรอเอเบิล จะต้องใช้ฟังก์ชันหลีกหมายเลข 5 เพื่อแปลงตำแหน่งของตัวชี้ตำแหน่งจากระบบพิกัดของอุปกรณ์เป็นระบบพิกัดโลก ซึ่งฟังก์ชันนี้ต้องการใช้สโตร์ด้วย ถ้าต้องการตามการเคลื่อนไหวของตัวชี้ตำแหน่งตลอดเวลา ก็จะต้องสร้างและลบสโตร์ตลอดเวลาเช่นกัน ซึ่งจะทำให้โปรแกรมทำงานช้าลง บางครั้งอาจทำให้การปรับภาพไม่ทันกับการเคลื่อนไหวของตัวชี้ตำแหน่ง หรือในบางครั้งอาจทำให้โปรแกรมตีความอีเวนต์ผิดพลาดได้

ผลของการพัฒนาโปรแกรมโดยใช้ซันฟิกส์รุ่น 2.0

การพัฒนาโปรแกรมในช่วงแรกได้เลือกใช้เวิร์กสเตชันแบบเอกซ์ ทูล ซึ่งเป็นแบบที่มีกระบวนการของฟิกส์มอนิเตอร์ที่ทำหน้าที่ติดต่อและดูแลการทำงานกับเอกซ์วินโดว์ทั้งหมด และดังที่ได้กล่าวมาแล้วว่า การแสดงภาพกราฟิกระดับสูงของซันฟิกส์รุ่น 2.0 ทำได้ค่อนข้างดีและใช้ง่าย ดังนั้นการพัฒนาโปรแกรมในช่วงแรกจึงสามารถให้ความสำคัญในการพัฒนาวิธีการทำงานต่างๆ ของโปรแกรม (เช่น วิธีการสร้างวัตถุ การแก้ไขวัตถุ เป็นต้น) ได้อย่างเต็มที่ และเนื่องจากการแสดงผลกราฟิกทำได้ง่าย การพัฒนาโปรแกรมจึงสามารถใช้วิธีทดลองให้เห็นภาพจริง และทำการปรับแก้ข้อมูลต่างๆ ในเวลานั้นได้ การพัฒนาโปรแกรมจึงมีลักษณะการพัฒนาจากส่วนในไปส่วนนอก คือ พัฒนากระบวนการภายในก่อน แล้วจึงพัฒนาส่วนรับข้อมูล ส่วนแสดงผลและส่วนตัวประสานผู้ใช้ภายหลัง แต่เมื่อได้พัฒนาถึงส่วนรับข้อมูลของโปรแกรมได้พบปัญหาในส่วนรับข้อมูลของซันฟิกส์รุ่น 2.0 ดังที่ได้กล่าวมาแล้ว จึงทำให้ต้องมีการเปลี่ยนเวิร์กสเตชันจากชนิดเอกซ์ทูลมาเป็นชนิดเอกซ์ ดรอเอเบิลแทน เพื่อใช้ส่วนรับข้อมูลจากเอกซ์วินโดว์

การพัฒนาโปรแกรมโดยใช้เวิร์กสเตชันแบบเอกซ์ ดรอเอเบิล จะมีลักษณะการพัฒนาจากส่วนนอกมาสู่ส่วนใน คือ ต้องพัฒนาส่วนการจัดการกับส่วนรับข้อมูลและส่วนแสดงผลบางส่วนก่อนจึงสามารถพัฒนากระบวนการภายในโปรแกรมได้ ถ้ามีการพัฒนากระบวนการภายในก่อนจะไม่สามารถรู้ผลการพัฒนาได้เลย เนื่องจากยังไม่ได้กำหนดส่วนการแสดงผลให้ทำงานสอด

คล้องกับเอกซ์วินโดว์ นอกจากนี้เมื่อเปลี่ยนจากเอกซ์ ทูลมาเป็นเอกซ์ ดรอเอเบิล ยังทำให้ต้องมีการเปลี่ยนลักษณะของโปรแกรมจากเดิมที่เป็นแบบโครงสร้าง มาเป็นแบบที่ใช้อีเวนต์และโนติฟายเออร์ และในแบบเอกซ์ ดรอเอเบิลนี้ซันฟิกส์รุ่น 2.0 จะไม่มีการสร้างกระบวนการของฟิกส์มอนิเตอร์ ซึ่งทำให้งานการติดต่อและควบคุมระบบซึ่งฟิกส์มอนิเตอร์เคยทำอยู่ โปรแกรมต้องมาจัดการเอง ดังนั้นแทนที่จะสามารถให้ความสำคัญกับการพัฒนาการทำงานของโปรแกรมได้เต็มที่ ทำให้ต้องมาเสียเวลาพัฒนาในส่วนการติดต่อและควบคุมระบบ

เนื่องจากซันฟิกส์รุ่น 2.0 ขาดความยืดหยุ่นต่อการเปลี่ยนแปลง และไม่มีการจัดการข้อมูลที่ดี ดังนั้นในการพัฒนาโปรแกรมจะต้องเสียเวลาส่วนหนึ่งมาจัดการเรื่องเหล่านี้ และเนื่องจากซันฟิกส์รุ่น 2.0 มีชนิดของโครงสร้างข้อมูลจำนวนมาก แม้ว่าจะได้ทำการสร้างฟังก์ชันสำหรับการทำงานกับโครงสร้างข้อมูลชนิดหนึ่งแล้ว ก็ยังต้องสร้างฟังก์ชันใหม่สำหรับการทำงานแบบเดียวกัน เพื่อใช้กับโครงสร้างข้อมูลชนิดอื่นๆ ถึงแม้จะเป็นโครงสร้างข้อมูลของรูปทรงพื้นฐานในกลุ่มเดียวกันก็ตาม เช่น โพลีไลน์กับโพลีไลน์ที่กำหนดด้วยข้อมูล หรือกลุ่มของพื้นที่เติมเต็มที่กำหนดด้วยข้อมูลกับชุดของกลุ่มของพื้นที่เติมเต็มที่กำหนดด้วยข้อมูล เป็นต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย