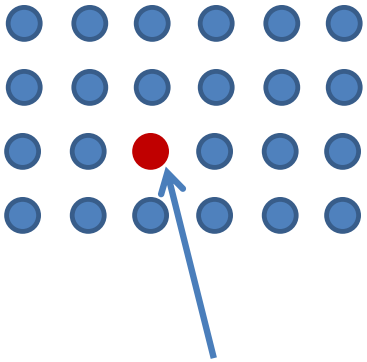
OpenGL และ GLUT

การแสดงผลในคอมพิวเตอร์

- จุดสี (pixel)
  - ตำแหน่งบนภาพ
- สี (color)



จุดสีที่ (3, 3) มีสีแดง

สี

- แสดงผลด้วย 3 ค่า
- RGB แดง เขียว น้ำเงิน
  - แดง (1,0,0); เขียว (0,1,0); น้ำเงิน (0,0,1)
  - ขาว (1,1,1); ดำ (0,0,0)
  - สีอื่นๆ ?

OpenGL

- State machine: เราเป็นคนบอก OpenGL ว่ากำลังจะทำอะไร
  - ตั้งคุณสมบัติต่างๆ ของ window
  - ตั้งค่าสีของสิ่งที่กำลังจะวาด
  - ตั้งคุณสมบัติอื่นๆ เช่น การฉายภาพ
- โปรแกรมแบบตอบสนอง: callbacks
  - คิดว่ามี action อะไรบ้าง แล้วจะตอบสนองกับ action นั้นอย่างไร

Callback

- glutDisplayFunc – ถูกเรียกเมื่อต้องการแสดงผล
- glutMouseFunc – ถูกเรียกเมื่อมีการกระทำกับ mouse
- glutKeyboardFunc – ถูกเรียกเมื่อมีการกด keyboard

รูปแบบการเขียนโปรแกรม

- OpenGL function ขึ้นต้นด้วย gl, ต่อด้วยตัวพิมพ์ใหญ่: glBegin, glPolygonMode
- ค่าคงที่: GL_2D, GL_RGB, …
- ประเภทข้อมูล: GLbyte, GLfloat, …

GLUT

- Utility toolkit
  - แสดงผล
  - การติดต่อกับอุปกรณ์
  - GLUT functions: glutInitWindowSize, glutIdleFunc, …
  - GLUT ค่าคงที่: GLUT_RIGHT_BUTTON

การเริ่มต้น

- #include <GL/glut.h>
  - นำเข้า library ต่างๆ ของ windows และ OpenGL
- ***glutInit*** (int * argcp, char **argv)
  - เริ่มต้น GLUT library ตาม option
- ***glutInitWindowSize*** (int width, int height)
- ***glutInitWindowPosition*** (int x, int y)
- ***glutInitDisplayMode*** (unsigned int mode)
  - GLUT_RGBA | GLUT_DEPTH | GLUT_DOUBLE, etc…
  - ส่งค่าผ่าน logic OR ของค่าคงที่
- ***glutCreateWindow*** (char *window_name)

เริ่มสร้าง window

- glutInitWindowPosition (350, 100);
  - วางตำแหน่ง window ตรงไหน
- glutInitWindowSize (winWid, winHght);
  - ขนาดของ window
- glutCreateWindow (“Triangle Program");
- glClearColor (1.0, 1.0, 1.0, 0.0)
  - สีพื้นหลัง
  - สามตัวแรกคือค่า RGB
  - ตัวที่สี่ใช้ร่วมกับการสร้างภาพที่มีค่าโปร่งแสง (ยังไม่ใช้)

การฉาย (Projection)

- glMatrixMode (GL_PROJECTION);
  - เข้า mode การตั้งค่าการฉายด้วยเมตริกซ์
- gluOrtho2D (0.0, winWth, 0.0,winHght);
  - ตั้งค่าการฉายภาพแบบ orthographic (อธิบายภายหลัง)

GLUT Callback

- ***glutDisplayFunc*** (void (*func) (void))
- ***glutReshapeFunc*** (void (*func) (int width, int height))
- ***glutKeyboardFunc***(void (*func) (unsigned char key, int x, int y))
  - ตำแหน่ง mouse (x, y) เมื่อมีการกดปุ่มบน keyboard
- ***glutMouseFunc*** (void (*func) (int button, int state, int x, int y))
  - ปุ่ม: GLUT_LEFT_BUTTON, GLUT_MIDDLE_BUTTON, GLUT_RIGHT_BUTTON
  - สถานะ: GLUT_UP , GLUT_DOWN
  - ตำแหน่ง (x, y): อ้างอิงกับตำแหน่ง window

GLUT Callback

- ***glutMotionFunc*** (void (*func) (int x, int y))
  - การเคลื่อนที่ของ mouse เมื่อกดปุ่มไว้
- ***glutPassiveMotionFunc*** (void (*func) (int width, int height))
  - การเคลื่อนที่ของ mouse เมื่อไม่มีการกดปุ่ม
- ***glutIdleFunc***(void (*func) (void))
  - เรียกเมื่อไม่มี event เหลือให้ทำ
  - ส่งค่า NULL ถ้าไม่ต้องการใช้
- ***glutTimerFunc*** (unsigned int msecs, void (*func) (int value), value))
  - เรียกทุกๆ msecs มิลลิวินาที
  - เรียก func ที่รับค่า value
  - ใช้ได้มากกว่า 1 timer

GLUT วงวนหลัก

- ***glutMainLoop*** (void)
  - เริ่มการทำงานของ GLUT
  - จัดการเรียก callback ให้ตาม event ที่เกิด
  - เรียกครั้งเดียวพอ

GLUT พื้นฐาน

โครงสร้างโปรแกรม

1. ตั้งค่าและสร้าง window (GLUT)
2. ตั้งค่าเริ่มต้น OpenGL (Optional)
3. ลงทะเบียน callback functions (GLUT)
   - Render
   - Resize
   - Input: keyboard, mouse, etc
4. เริ่มวงวนหลัก (GLUT)

ตัวอย่างโปรแกรม

```
#include <GL/glut.h>
#include <GL/gl.h>

Void main(int argc, char** argv)
{
    int mode = GLUT_RGB|GLUT_SINGLE;
    glutInitDisplayMode(mode);
    glutInitWindowSize(500,500);
    glutCreateWindow(argv[0]);
    init();
    glutDisplayFunc(display);
    glutKeyboardFunc(key);
    glutMainLoop();
}
```

จุดยอด (Vertices)

- ***glVertex2s*** *(200, -150);*
  - จุดพิกัด 2D ประเภท short
- ***glVertex3i*** *(200, -150, 40);*
  - จุดพิกัด 2D ประเภท integer
- ***GLdouble*** *dpoint[3] = {200.0, -150.5, 40.0};*
  glVertex3dv *(dpoint);*
  - จุดพิกัด 3D ประเภท double

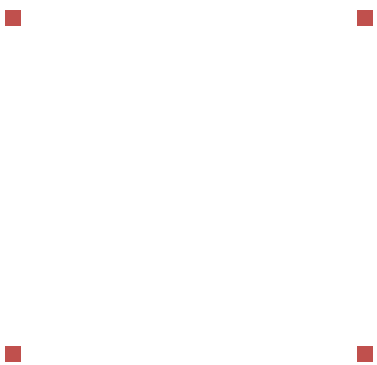
จุด, เส้น, รูปหลายด้าน (Polygons)

- ***glBegin(mode)*** และ ***glEnd( )*** ครอบการกำหนดจุดของสิ่งที่จะสร้าง

- ***mode*** มีดังนี้:
  - GL_POINTS
  - GL_LINES
  - GL_POLYGON
  - GL_LINE_STRIP
  - GL_TRIANGLE_STRIP
  - GL_TRIANGLES
  - GL_QUADS
  - GL_LINE_LOOP
  - GL_QUAD_STRIP
  - GL_TRIANGLE_FAN

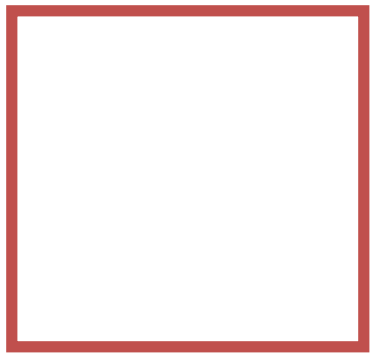
จุด

```
glBegin(GL_POINTS);
glVertex2i( 0, 0 );
glVertex2i( 0, 1 );
glVertex2i( 1, 0 );
glVertex2i( 1, 1);
glEnd( );
```



เส้นรอบรูปของรูปหลายเหลี่ยม (Polyline)

```
glBegin(GL_LINE_LOOP);
glVertex2i( 0, 0 );
glVertex2i( 0, 1 );
glVertex2i( 1, 1 );
glVertex2i( 1, 0 );
glEnd( );
```

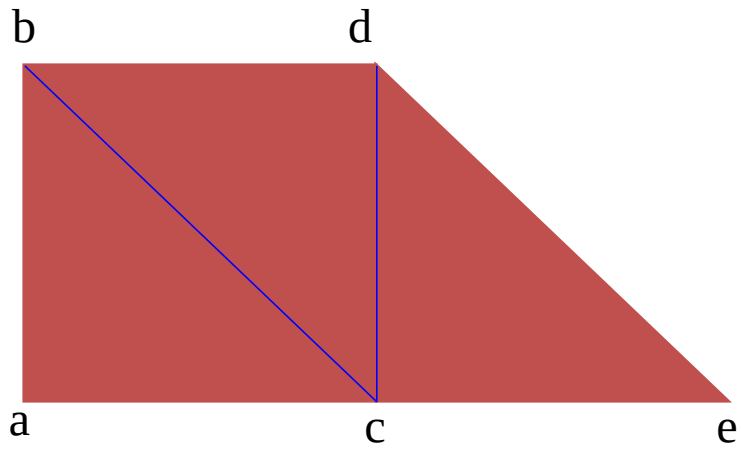


รูปหลายเหลี่ยม

```
glBegin(GL_POLYGON);
glVertex2i( 0, 0 );
glVertex2i( 0, 1 );
glVertex2i( 1, 1 );
glVertex2i( 1, 0 );
glEnd( );
```

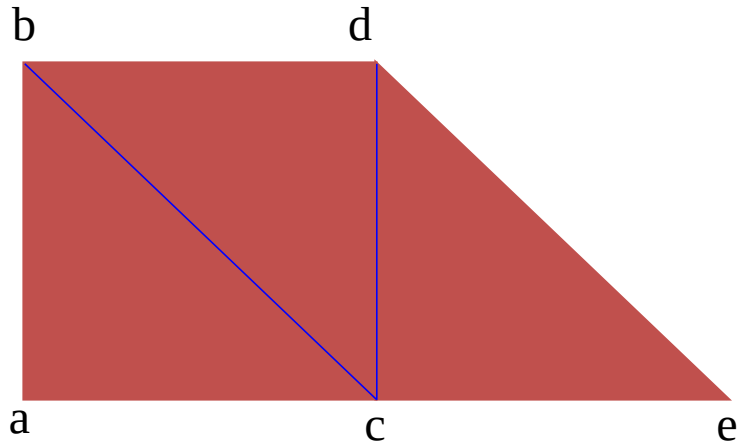
สามเหลี่ยม

```
glBegin(GL_TRIANGLES);
glVertex2i( 0, 0 ); // a
glVertex2i( 0, 1 ); // b
glVertex2i( 1, 0 ); // c
glVertex2i( 0, 1 ); // b
glVertex2i( 1, 0 ); // c
glVertex2i( 1, 1 ); // d
glVertex2i( 1, 0 ); // c
glVertex2i( 1, 1 ); // d
glVertex2i( 2, 0 ); // e
glEnd( );
```



สามเหลี่ยมแบบจุดต่อเนื่อง (triangle strip)

```
glBegin(GL_TRIANGLE_STRIP);
glVertex2i( 0, 0 ); // a
glVertex2i( 0, 1 ); // b
glVertex2i( 1, 0 ); // c
glVertex2i( 1, 1 ); // d
glVertex2i( 2, 0 ); // e
glEnd( );
```



คุณลักษณะ

- จุด
  - ขนาด : ***glPointSize****(2.0);*
  - สี : ***glColor3f*** *(0.0, 0.0, 1.0);*
- เส้น
  - ความหนา : ***glLineWidth****(2.0);*
  - สี : ***glColor3f*** *(0.0, 0.0, 1.0);*
- พื้นผิว
  - ให้สีด้านหน้าหรือด้านหลัง : ***GL_FRONT, GL_BACK, GL_FRONT_AND_BACK***
  - สี : ***glColor3f*** *(0.0, 0.0, 1.0);*